SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 27 de Outubro de 2010

Assinatura:_____

Sistema de identificação de superfícies navegáveis baseado em visão computacional e redes neurais artificiais

Patrick Yuri Shinzato

Orientador: ***Prof. Dr. Denis Fernando Wolf***

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências - Ciências de Computação e Matemática Computacional.

USP – São Carlos
Outubro/2010

Sistema de identificação de superfícies navegáveis baseado em visão computacional e redes neurais artificiais

Patrick Yuri Shinzato

Agradecimentos

Agradeço a todos que de forma direta ou indireta participaram da execução deste trabalho. Neste sentido, agradeço ao meu orientador, o professor Denis Fernando Wolf, pelos ensinamentos, pela oportunidade da realização deste mestrado, do desenvolvimento deste projeto e principalmente pela amizade e confiança.

Agradeço a toda a minha família pelo apoio. Agradeço aos meus pais, Cristina e Silvio Shinzato, pelo apoio e incentivo para concluir a faculdade e seguir a carreira acadêmica. Agradeço a minha irmã, Marjolly, pela convivência e pela ajuda na revisão ortográfica deste documento.

Um agradecimento especial a todos os meus amigos do laboratório que me aguentam pois pego no pé da maioria deles, obrigado pela ajuda e pela convivência. Obrigado Alberto pela ajuda e companherismo aqui e até mesmo no Chile. Obrigado Pessin pelo companherismo e por também ter revisado este documento. Obrigado também a todos os novos amigos de São Carlos pelas festas, churrascos, bagunças, e viagens. E claro, a todos os amigos antigos de Campo Grande.

Resumo

A Navegação autônoma é um dos problemas fundamentais da robótica móvel. Para um robô executar esta tarefa, é necessário determinar a região segura para a navegação. Este trabalho propõe um sistema de identificação de superfícies navegáveis baseado em visão computacional utilizando redes neurais artificiais. Mais especificamente, é realizado um estudo sobre a utilização de diferentes atributos de imagem, como descritores estatísticos e elementos de espaços de cores, para serem utilizados como entrada das redes neurais artificiais que tem como tarefa a identificação de superfícies navegáveis.

O sistema desenvolvido utiliza resultados de classificação de múltiplas configurações de redes neurais artificiais, onde a principal diferença entre elas é o conjunto de atributos de imagem utilizados como entrada. Essa combinação de diversas classificações foi realizada visando maior robustez e melhor desempenho na identificação de vias em diferentes cenários.

Abstract

Autonomous navigation is a fundamental problem in mobile robotics. In order to perform this task, a robot must identify the areas where it can navigate safely. This dissertation proposes a navigable terrain identification system based on computer vision and neural networks. More specifically, it is presented a study of image attributes, such as statistical decriptors and elements of different color spaces, that are used as neural neworks inputs for the navigable surfaces identification.

The system developed combines the classification results of multiple neural networks topologies with different image attributes. This combination of classification results allows for improved efficient and robustenes in different scenarios.

Sumário

Agradecimentos **i**

Resumo **iii**

Abstract **v**

Lista de Figuras **ix**

Lista de Tabelas **xiii**

1 Introdução **1**
1.1 Motivação 2
1.2 Objetivo 3
1.3 Estrutura da Dissertação 4

2 Trabalhos Relacionados **5**
2.1 NAVLAB 6
2.2 Modelo Adaptativo Baseado em Cor 8
2.3 Combinação de Laser, Cor e Textura 9
2.4 Modelo Adaptativo Utilizando RNA 10
2.5 Sistema Visual Para Controle de Velocidade 11
2.6 Considerações 12

3 Metodologia **13**
3.1 Gerador de Atributos 14
3.1.1 Método de Segmentação Baseada em Blocos 15
3.1.2 Atributos de Imagem 16

3.1.2.1 Cor 17
3.1.2.2 Textura 17
3.1.2.3 Histograma 18
3.1.2.4 Descritores Estatísticos 19
3.1.3 Considerações 19
3.2 Classificador 20
3.2.1 Aprendizado de Máquina 21
3.2.2 Métodos de Avaliação de Desempenho do Classificador 23
3.2.3 Modelo Final do Classificador e Considerações 26
3.3 Navegação 30
3.3.1 Navegação Autônoma Baseada em Visão Utilizando Controle Baseado em Autômato Finito Determinístico 30
3.3.2 Navegação Autônoma em Ambientes Urbanos Baseado em Casamento de Modelos 32
3.4 Considerações 34

4 Experimentos e Resultados **35**
4.1 Seleção de Atributos 36
4.1.1 Seleção de Atributos Baseado na Saliência 36
4.1.2 Análise de Desempenho das RNA com Diferentes Atributos de Entrada 40
4.2 Método de Avaliação 46
4.3 Sistema de Identificação 50
4.4 Identificação da Linha do Horizonte 55
4.5 Outros Resultados 57
4.6 Considerações 58

5 Conclusão **59**
5.1 Trabalhos Futuros 60

Referências Bibliográficas **61**

A Publicações Obtidas Como Resultado Desse Trabalho **65**

B Lista de Atributos Avaliados **67**

C Cenários Utilizados **69**

Lista de Figuras

1.1 Robôs móveis desenvolvidos para atuarem em ambientes inacessíveis ao homem. 2
1.2 Estatísticas sobre causas de acidentes no Trânsito. Imagem adaptada de (ITARDA, 2010). 3
1.3 Plataformas de testes da USP. 4

2.1 Plataformas de testes NAVLAB desde 1987 até 2010. As imagens mostram 2 dos veículos utilizados pelos pesquisadores da Carnegie Mellon ao longo dos últimos anos. 6
2.2 Diagramas de blocos dos projetos desenvolvidos sobre a plataforma NAVLAB. 7
2.3 Resultados de (Tan *et al.*, 2006) - Exemplo de rua com imagem não homogênea. Neste caso de exemplo, a rua possui influência de sombra a partir de um determinado momento. Mesmo assim, o sistema identifica a rua de maneira eficiente. 8
2.4 Resultados de (Rasmussen, 2002). A bolinhas pintadas de verde representam a região que o sistema considera navegável. As pintadas de vermelho representam a região não navegável. Observe que a fusão da informação visual com a informação do laser melhora consideravelmente a classificação. 9
2.5 Sistema desenvolvido por (Foedisch, 2004). A imagem (a) mostra um exemplo da forma de geração de padrões de treinamento e a imagem (b) mostra o diagrama do sistema na etapa de treinamento e execução. 10
2.6 Execução do algoritmo de processamento de imagem desenvolvido por (Thrun *et al.*, 2006). Baseado no padrões encontrados na região considerada navegável pelo laser (quadrilátero azul), o sistema identifica a região navegável na imagem. 11
2.7 Rápida adaptação de Stanley. Utiliza o laser para definir os padrões de cor, faz com que o sistema se adapte mesmo quando a cor da superfície muda drasticamente. 12

3.1 Diagrama do Sistema: mostra cada uma das etapas detalhando os dados necessários para cada fase. Os blocos pintados de vermelho no mapa são os elementos considerados navegáveis. 14
3.2 Segmentação de imagem baseada em blocos. 16
3.3 Representações gráficas dos espaços de cores utilizados. Imagens retiradas de (Couleur, 2010). 18
3.4 Diagrama que descreve o funcionamento de um classificador. O classificador utiliza os valores dos atributos de um elemento de imagem para determinar qual classe ele pertence. 21
3.5 Estrutura básica de um neurônio. 22
3.6 Representações gráficas das topologias de RNA. 23
3.7 Funções aplicadas como pesos nos valores do histograma. 25
3.8 Topologia da RNA utilizada neste trabalho como classificador. Os atributos do elemento são as entradas da RNA que retorna um resultado pelo neurônio da camada de saída. 27
3.9 Inicialização do Sistema: o quadrado vermelho engloba os blocos de pixel considerados navegáveis pelo usuário, e o quadrado azul os considerados não navegáveis. 27
3.10 Resultados obtidos da classificação de algumas imagens. As imagens a esquerda são as imagens obtidas pela câmera. As imagens a direita mostram o resultado da classificação onde quanto mais branco maior a certeza de navegabilidade. 28
3.11 Diagrama do Sistema de Identificação Baseado em uma RNA. 29
3.12 Diagrama do Sistema de Identificação Baseado em múltiplas RNAs. 29
3.13 Experimento utilizando o Sistema de Identificação de Superfícies Navegáveis proposto. Imagens retiradas de (Sales *et al.*, 2010). 30
3.14 Descrição do experimento utilizando o Sistema de Identificação de Superfícies Navegáveis proposto. Imagens retiradas de (Sales *et al.*, 2010). 31
3.15 Comparação de imagem original e do mapa de navegabilidade de um experimento utilizando o Sistema de Identificação de Superfícies Navegáveis proposto. Imagens retiradas de (Sales *et al.*, 2010). 31
3.16 Experimento em ambiente externo utilizando o Sistema de Identificação de Superfícies Navegáveis. Imagens retiradas de (Souza *et al.*, 2011). 32
3.17 Experimento em ambiente externo em outro tipo de superfície. Mostra como o sistema é robusto, uma vez que ele consegue identificar a região navegável mesmo sendo diferente do asfalto. 33

4.1 EQM das RNA-1, RNA-2 e RNA-3 para o Cenário 1, onde a linha vermelha é a média do EQM das imagens, o azul mostra o melhor e pior EQM e a linha preto é a mediana dentre os EQM obtidos. 38
4.2 EQM das RNA-1, RNA-2 e RNA-3 para o Cenário 2, onde a linha vermelha é a média do EQM das imagens, o azul mostra o melhor e pior EQM e a linha preto é a mediana dentre os EQM obtidos. 39

4.3 Resultados das RNA que obtiveram o melhor e pior desempenho. As imagens (c) e (f) mostram que a RNA-3 não diferencia os blocos navegáveis dos não-navegáveis, pois quase todos os blocos receberam a nota $0,5$, o que significa que a RNA possui total incerteza sobre a classe a qual o bloco pertence. As imagens (b) e (e) mostram os bons resultados da RNA-1, pois grande parte da região navegável foi classificada com um grau de certeza muito elevado por isso os blocos pintados de branco. 40
4.4 Resultados nos cenários 1 e 2 obtidos da RNA do grupo T2 com 10 atributos de imagem. 43
4.5 Resultados nos cenários 5 e 6 obtidos da RNA do grupo T2 com 10 atributos de imagem. 44
4.6 Resultados nos cenários 7 e 8 obtidos da RNA do grupo T2 com 10 atributos de imagem. 45
4.7 Experimento no Cenário 1. Diferença entre as avaliações EQM e FIT. Pode-se ver que o EQM indica uma convergência enquanto o FIT ainda mostra uma melhoria no aprendizado entre os ciclos 1000 e 1500. 47
4.8 Experimento no Cenário 1. Comparação dos resultados entre os ciclos 1000 e 1500. Pode-se ver que a certeza sobre a não navegabilidade dos blocos superiores aumenta conforme a RNA é treinada. 48
4.9 Experimento no Cenário 2. Diferença entre as avaliações EQM e FIT. Pode-se ver que o EQM indica uma convergência enquanto o FIT ainda mostra uma melhoria no aprendizado. 49
4.10 Experimento no Cenário 2. Comparação dos resultados entre os ciclos 500 e 1000 da mesma RNA. 49
4.11 Comportamento semelhante dos resultados EQM e FIT. 50
4.12 Exemplo de resultado bom: Resultados sobre o Cenário 1 do sistema treinado também com padrões do Cenário 1. O valor do FIT é $0,94$. As imagens (a)-(e) mostram as imagens originais. As imagens (f)-(j) mostram o resultado das classificações. As imagens (k)-(o) mostram o resultado sobre a imagem real, onde quanto mais certeza sobre a navegabilidade, mais vermelho é o bloco. Os blocos considerados não navegáveis mantém sua cor original. 51
4.13 Exemplo de resultado aceitável: Resultados sobre o Cenário 7 do sistema treinado com padrões do Cenário 5. O valor do FIT é $0,88$. As imagens (a)-(e) mostram as imagens originais. As imagens (f)-(j) mostram o resultado das classificações. As imagens (k)-(o) mostram o resultado sobre a imagem real, onde quanto mais certeza sobre a navegabilidade, mais vermelho é o bloco. Os blocos considerados não navegáveis mantém sua cor original. 52
4.14 Resultados sobre o Cenário 1 do sistema treinado com padrões do Cenário 7. O valor do FIT é $0,75$. 53
4.15 Resultados sobre o Cenário 4 do sistema treinado com padrões do Cenário 7. O valor do FIT é $0,72$. As imagens (k)-(o) mostram o resultado sobre a imagem real, onde quanto mais certeza sobre a navegabilidade, mais vermelho é o bloco. 54
4.16 Selecionando região de interesse do sistema de identificação. 55

4.17 Resultado mostrando a estimativa da linha do horizonte na imagem em quatro cenários distintos. 56
4.18 Experimentos em diferentes cenários. 57

C.1 Cenário 1 70
C.2 Cenário 2 71
C.3 Cenário 3 72
C.4 Cenário 4 73
C.5 Cenário 5 74
C.6 Cenário 6 75
C.7 Cenário 7 76
C.8 Cenário 8 77

Lista de Tabelas

3.1 Pesos utilizados para calcular o FIT. Cada peso é multiplicado por um determinado *bean* do histograma. 25

4.1 Resultados do Método de Seleção por Saliências. Cada coluna representa um cenário e mostra uma nota entre 0 e 1 para cada atributo. Quanto maior essa nota maior a sua importância na classificação por RNA. Os resultados maiores que 0,5 estão destacados em cinza, onde quanto maior a nota mais escuro é a célula. A tabela está ordenada pela coluna "Média". 37

4.2 Atributos candidatos aos melhores desempenhos. Três listas de atributos onde cada uma foi gerada a partir dos resultados sobre cenários em diferentes condições de iluminação. 41

4.3 Resultados dos EQM para seleção de atributos que serão utilizados no sistema. Para cada cenário a coluna "M" mostra o melhor desempenho alcançado e a coluna "F" o desempenho da RNA no último ciclo de treinamento. As células com erros menores que $0,03$ foram pintados de cinza escuro, as células com EQM menor que $0,05$ foram pintadas de cinza claro. 42

4.4 Taxa de acerto da RNA para os ciclos 1000 e 1500 no Cenário 1. 47

4.5 Resultados do sistema para todos os cenários apresentados. Cada coluna representa o FIT obtido em padrões de um cenário. Cada linha mostra a origem dos padrões com que o sistema foi treinado. 51

4.6 Comparação do desempenho do sistema utilizando detecção do horizonte ou não. 57

CAPÍTULO

1

Introdução

Um dos principais objetivos da robótica móvel é o desenvolvimento de robôs móveis autônomos que sejam capazes de interagir com um ambiente real, aprendendo e tomando decisões corretas para que suas tarefas sejam executadas com êxito, mesmo em ambientes dinâmicos e não-estruturados.

Dentre as diversas aplicações de robôs móveis autônomos, destaca-se o reconhecimento de locais inacessíveis ao homem ou que representem risco à vida. Como exemplo, pode-se citar o robô Sojourner (Figura 1.1(a)) da NASA que explora o planeta Marte (NASA, 2009), e o robô Groundhog (Figura 1.1(b)) desenvolvido na Universidade Carnegie Mellon, que realiza explorações em minas abandonadas com possíveis gases tóxicos e alto risco de desabamento (Thrun *et al.*, 2005). Além de robôs terrestres, existem projetos que utilizam plataformas robóticas para ambientes submarinos (Williams e Mahon, 2004) e aéreos (CarnegieMellon, 2010) (Stanford, 2010).

Dentre as linhas de pesquisa exploradas na robótica móvel, as funções relativas a mapeamento, localização e navegação tem grande importância no desenvolvimento de robôs móveis autônomos. Normalmente essas capacidades básicas são utilizadas como pré-requisito para que outras tarefas mais complexas possam ser executadas. Com relação a pesquisa em sistemas de navegação, grande parte dela foi direcionada para atuação dos robôs em ambientes altamente estruturados como no interior de casas, onde o foco principal do algoritmo de controle é o desvio de obstáculos. A navegação em ambientes externos, como vias públicas e terrenos desconhe-

(a) Sojouner (b) Groundhog

Figura 1.1: Robôs móveis desenvolvidos para atuarem em ambientes inacessíveis ao homem.

cidos, consiste em um problema muito mais complexo e vem recebendo grande atenção nos últimos anos. Além do desvio de obstáculos, é necessário que o robô identifique a região onde pode navegar. A irregularidade do terreno e a dinâmica do ambiente são alguns dos fatores que tornam a navegação de um robô mais difícil nesses ambientes. Tais fatores exigem algoritmos sofisticados que são capazes de interpretar os dados obtidos pelos sensores nas mais diferentes situações que esse ambiente pode apresentar.

1.1 Motivação

Uma aplicação direta da tecnologia de navegação em ambientes externos é o desenvolvimento de veículos autônomos inteligentes. Segundo a Associação Nacional de Transportes Públicos (ANTP, 2010), os custos anuais com acidentes de trânsito são da ordem de R$ 28 bilhões. Com base na média entre os anos de 2003 e 2006, cerca de 34 mil pessoas morrem no trânsito brasileiro por ano, além de algumas centenas de milhares que ficam feridos ou adquirem deficiências físicas temporárias ou permanentes. Na cidade de São Paulo ocorre um acidente a cada 3,2 minutos. A cada 7 horas morre um pedestre atropelado (ANTP, 2010). Estima-se que a causa de 90% desses acidentes é erro humano, imprudência, distração do motorista, ou incapacidade do mesmo em visualizar toda área ao redor do veículo. O desenvolvimento de veículos autônomos poderá contribuir na redução desse número de acidentes no trânsito. De acordo com (ITARDA, 2010), grande parte dos acidentes é causada por falhas do motorista como mostra a Figura 1.2. Essa redução também contribuirá na diminuição de engarrafamentos nas grandes cidades.

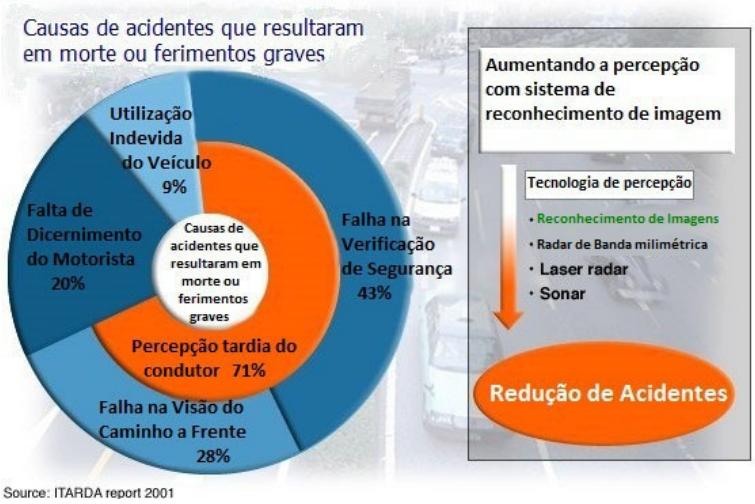


Figura 1.2: Estatísticas sobre causas de acidentes no Trânsito. Imagem adaptada de (ITARDA, 2010).

Outra motivação para o desenvolvimento de um veículo autônomo é o aumento de melhoria da mobilidade de portadores de necessidades especiais e de idosos. Segundo o IBGE (IBGE, 2009), no ano 2000, 11% da população brasileira tinha mais de 55 anos. Em 2050, esse percentual atingirá a cifra de 32,1%, aproximadamente 75 milhões de pessoas. Dirigir em grandes cidades e rodovias onde o ambiente é dinâmico pode ser extremamente perigoso para pessoas idosas, já que existe uma relação entre idade e diminuição de reflexos. Por isso, ao menor sinal de limitação, uma pessoa não tem sua carteira de habilitação renovada. Essa falta de mobilidade pode provocar isolamento e falta de socialização do indivíduo, como também o surgimento de depressão e outras doenças.

1.2 Objetivo

O objetivo dessa dissertação é o desenvolvimento de um sistema de identificação de superfícies navegáveis baseado em visão computacional. Em outras palavras, o sistema deve identificar uma região navegável utilizando métodos de visão computacional e um método de inteligência artificial (IA). Mais especificamente, o método de IA utilizado neste trabalho é um método de aprendizado de máquina supervisionado conhecido por redes neurais artificiais. Esse sistema pode ser incorporado a um sistema de navegação de um robô móvel, seja ele um carro ou uma plataforma robótica de menor porte.

Todo o trabalho deste projeto de mestrado faz parte de um projeto mais amplo que visa o desenvolvimento de veículos autônomos inteligentes para ambientes externos. Este projeto

está em desenvolvimento no Instituto de Ciências Matemáticas e de Computação (ICMC[1]) e consiste em uma parceria entre o INCT-SEC[2] (Instituto Nacional de Ciência e Tecnologia em Sistemas Embarcados Críticos) - que dispõe de um veículo elétrico (Figura 1.3(a)) - e o projeto SENA (Sistema Embarcado para Navegação Autônoma) da Escola de Engenharia de São Carlos (EESC), que possui um veículo doado pela FIAT para os testes experimentais (Figura 1.3(b)).

(a) Veículo elétrico do ICMC (b) Veículo do projeto SENA

Figura 1.3: Plataformas de testes da USP.

1.3 Estrutura da Dissertação

Esta dissertação está organizada da seguinte forma:

- Capítulo 2: Descreve trabalhos relacionados que utilizam visão computacional para realizar a navegação de robôs.
- Capítulo 3: Apresenta todo o sistema de identificação de superfícies navegáveis, desde o pré-processamento de imagem até o uso de redes neurais artificiais.
- Capítulo 4: Descreve vários experimentos realizados e discute os resultados obtidos.
- Capítulo 5: Apresenta a conclusão deste trabalho.

[1] `http://www.icmc.usp.br`
[2] `http://www.inct-sec.org`

CAPÍTULO

2

Trabalhos Relacionados

Existem inúmeros trabalhos de identificação de superfícies navegáveis na literatura. Trabalhos que se baseiam em cor e textura (Tan *et al.*, 2006) (Rasmussen, 2002) - que procuram regiões da imagem com a mesma característica -, ou baseados em forma (Khalifa *et al.*, 2009) (Jung e Kelber, 2005) - que procuram pelos limites da rua ou faixas de trânsito para detectar a forma trapezoidal da rua dentro da imagem. Como o objetivo deste trabalho é desenvolver um sistema de identificação visual de superfícies navegáveis em ambientes externos, essa dissertação se restringe a discutir projetos que também tiveram como foco o ambiente externo. Neste capítulo foram demonstrados projetos que utilizaram visão computacional para realizar a navegação ou para identificar a superfície navegável.

De acordo com (Bonin-Font *et al.*, 2008), a capacidade de navegar em ambientes externos estruturados (e.g. ruas em ambientes urbanos) baseada em visão pode ser chamada de *Road Following*. Sistemas de *Road Following* são sistemas com a habilidade de fazer um robô localizar e trafegar por uma rua. Além de se manter na região destinada a seu uso (rua), o sistema deve também fazer com que o robô evite obstáculos. Os trabalhos apresentados a seguir são ou fazem parte de sistemas de *Road Following*. Ao final deste capítulo, a Seção 2.6 traz um comentário geral dos projetos desenvolvidos e como este trabalho se encaixa no cenário apresentado.

2.1 NAVLAB

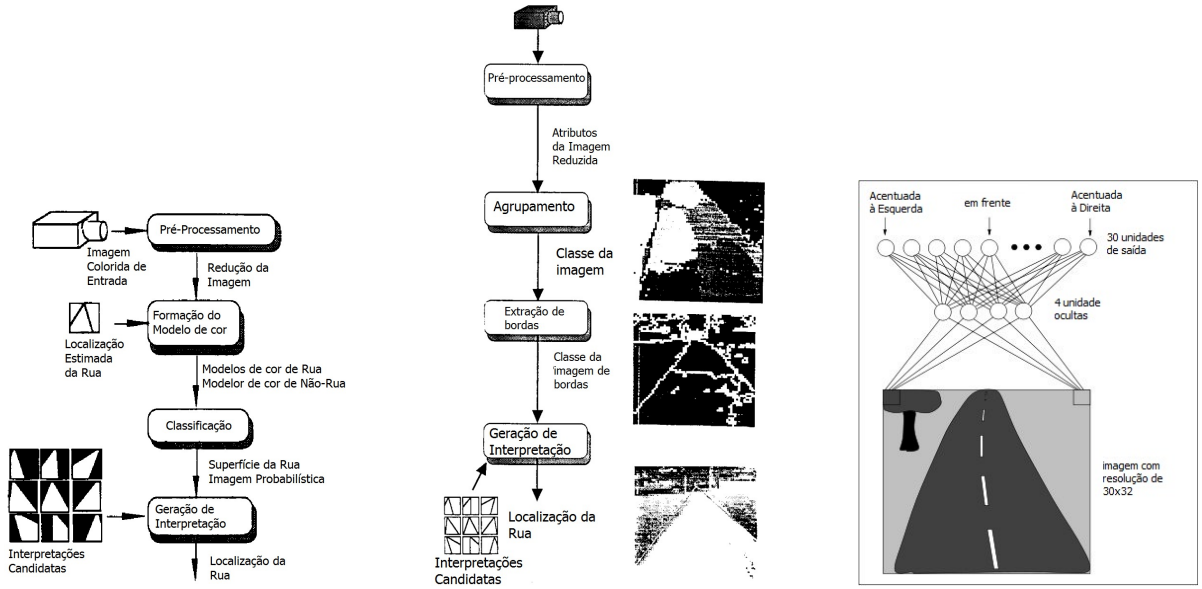
Um dos primeiros e mais notáveis projetos de *Road Following* foi desenvolvido em 1988 por (Thorpe *et al.*, 1988). Hoje, este projeto encontra-se na sua décima primeira versão. A Figura 2.1 mostra a primeira e a atual plataforma de desenvolvimento da Carnegie Mellon. Em sua primeira versão, o sistema de navegação do NAVLAB possuía três etapas: na primeira fase, uma classificação de cor e textura era executada definindo-se uma distribuição gaussiana para cada pixel do tipo *road* e *non-road*. Na segunda etapa, uma transformação de *Hough* seguida de um processo de votação era aplicado nos pixeis do tipo *road* para se obter o ponto de fuga e parâmetros de orientação. Por último, os pixeis eram novamente classificados de acordo com as bordas da rua. Essa nova classificação era usada para criar os modelos utilizados na próxima imagem, afim de manter o sistema adaptativo a mudanças das condições da rua.

(a) Navlab 1 (1987) (b) Navlab 11 (2010)

Figura 2.1: Plataformas de testes NAVLAB desde 1987 até 2010. As imagens mostram 2 dos veículos utilizados pelos pesquisadores da Carnegie Mellon ao longo dos últimos anos.

Posteriormente, a mesma plataforma - o NAVLAB -, foi utilizada no desenvolvimento de outros sistemas como o SCARF, o UNSCARF e o ALVINN. O sistema SCARF (*Supervised Classification Applied to Road Following*) (Crisman e Thorpe, 1993) identifica a rua calculando, para cada pixel, um valor de verossimilhança que descreve sua correspondência com as cores definidas como *road* e *non-road*. Esse sistema é rápido e robusto desde que ocorram mudanças suaves no ambiente como por exemplo as variações de iluminação durante o dia. O sistema UNSCARF (*UNSupervised Classification Applied to Road Following*) (Crisman e Thorpe, 1991) é mais lento que o SCARF, mas provê bons resultados mesmo com mudanças

drásticas do ambiente. No UNSCARF não existe a necessidade de cores definidas para *road* e *non-road*, pois inicialmente várias classes são criadas, mas em poucas iterações, estas se associam formando duas classes principais. Ambos os sistemas foram desenvolvidos na mesma época (Crisman e Thorpe, 1988) como sistemas de visão de ambientes externos. A Figura 2.2(a) e (b) mostra o diagrama de blocos desses sistemas.



(a) Diagrama de blocos do SCARF. Imagem adaptada de (Crisman e Thorpe, 1993)

(b) Diagrama de blocos do UNSCARF. Imagem adaptada de (Crisman e Thorpe, 1991)

(c) Funcionamento da rede neural do Projeto ALVINN. Imagem adaptado de (Pomerleau, 1996)

Figura 2.2: Diagramas de blocos dos projetos desenvolvidos sobre a plataforma NAVLAB.

O projeto ALVINN (*Autonomous Land Vehicle in a Neural Network*), desenvolvido por (Pomerleau, 1996), utilizava redes neurais artificiais (RNA) e processamento de imagem para conduzir um veículo de maneira autônoma em diversos tipos de ambientes. Os pixeis das imagens capturadas por uma câmera montada na parte superior do veículo alimentavam uma rede neural do tipo *backpropagation* (Haykin, 1998), previamente treinada, que fornecia como resultado os comandos que deveriam ser diretamente aplicados à direção do veículo. A Figura 2.2(c) mostra seu funcionamento. De acordo com (Pomerleau, 1996), seu sistema pode processar 15 imagens por segundo depois que a RNA está treinada, entretanto para isso, é preciso uma redução da resolução da imagem seja reduzida para 30×32 pixeis. Além disso, é necessário que exista uma RNA treinada para cada situação específica de ruas como faixa única com acostamento de terra, faixa única ladeada por ciclovia, faixa dupla etc.

2.2 Modelo Adaptativo Baseado em Cor

O sistema de (Tan *et al.*, 2006) se baseia na idéia de separar a rua da paisagem usando histogramas de cores. Para a rua é usado um conjunto de histogramas e para a paisagem é usado apenas um histograma. Manter vários histogramas para a rua ajuda o algoritmo à detectá-la mesmo com variações da iluminação e presença de sombra, pois o sistema pode possuir um modelo para cada situação. O algoritmo cria o histograma de uma parte da imagem, da região bem à frente do veículo, assumindo que essa região da imagem faz parte da rua.

O algoritmo inicia com um histograma da primeira imagem. Para cada imagem subsequente um novo histograma é criado e pode ser adicionado ao conjunto de modelos de cores ou substituir um já existente no conjunto. Normalmente o histograma mais antigo é descartado, assim o algoritmo é capaz de se adaptar às variações de cor e textura da rua. Além da construção e atualização do conjunto de modelos de cor para a rua, é realizado um cálculo de probabilidade para determinar se um certo pixel faz parte ou não da rua. Após essas etapas obtêm-se um mapa visual probabilístico ilustrado na Figura 2.3.

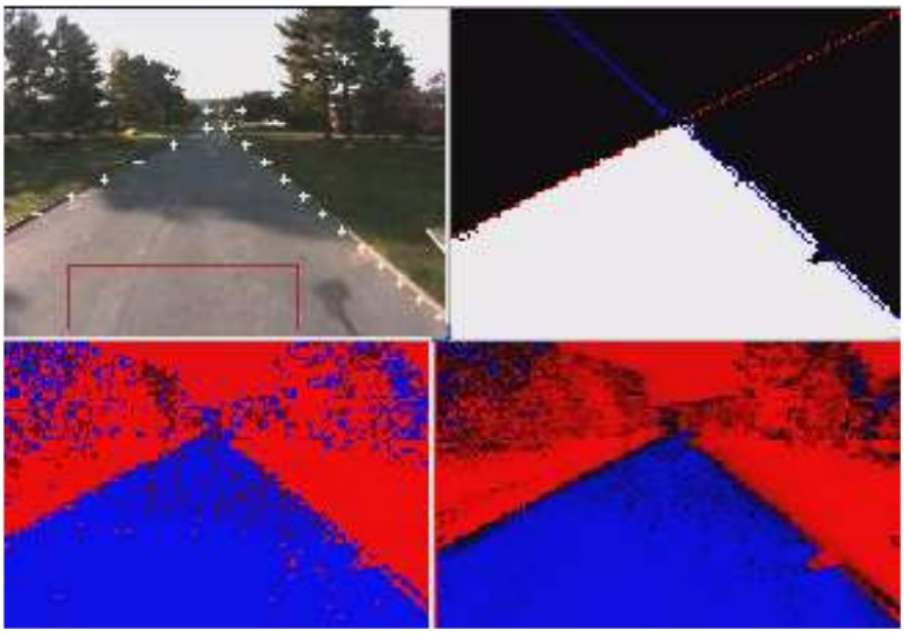
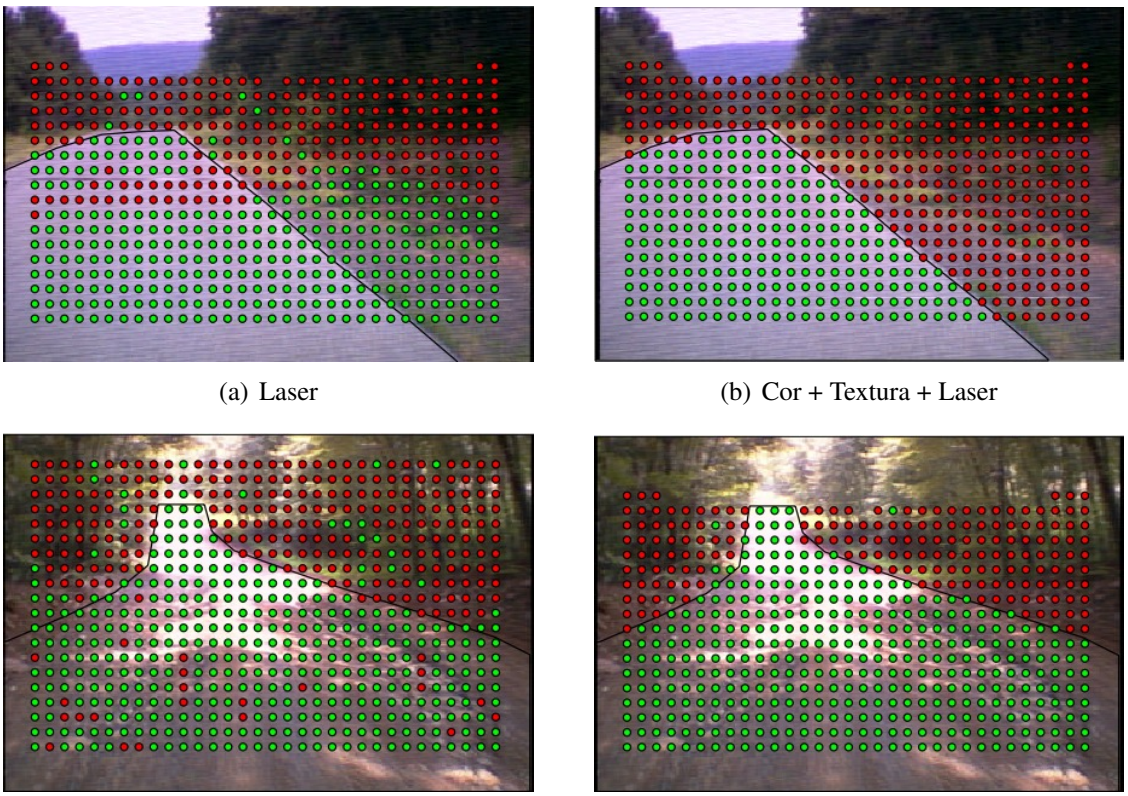


Figura 2.3: Resultados de (Tan *et al.*, 2006) - Exemplo de rua com imagem não homogênea. Neste caso de exemplo, a rua possui influência de sombra a partir de um determinado momento. Mesmo assim, o sistema identifica a rua de maneira eficiente.

2.3 Combinação de Laser, Cor e Textura

O sistema de (Rasmussen, 2002) combinou duas informações de profundidade - altura e homogeneidade - do laser com valores de cor e textura obtidas por uma câmera. Cada imagem foi dividida em 640 "centros de características", cada centro tem 10 pixeis de distância para outro centro, tanto na linha horizontal quanto na vertical da imagem. Para cada "centro de características", diferentes dimensões de sub-imagens foram examinadas, 7×7, 15×15 e 31×31 pixeis. Finalmente, para cada sub-imagem, foram calculadas 14 características, onde seis delas são dados dos histogramas, outros dois valores são referentes à textura, obtidos por meio de filtros de Gabor (Clark *et al.*, 1987) e os seis restantes são características obtidas a partir de informações do laser.



(a) Laser

(b) Cor + Textura + Laser

(c) Textura

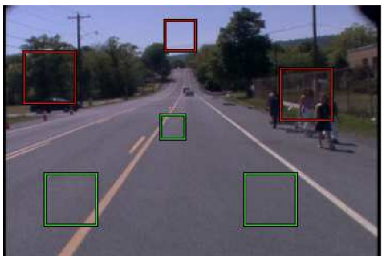
(d) Cor + Textura + Laser

Figura 2.4: Resultados de (Rasmussen, 2002). A bolinhas pintadas de verde representam a região que o sistema considera navegável. As pintadas de vermelho representam a região não navegável. Observe que a fusão da informação visual com a informação do laser melhora consideravelmente a classificação.

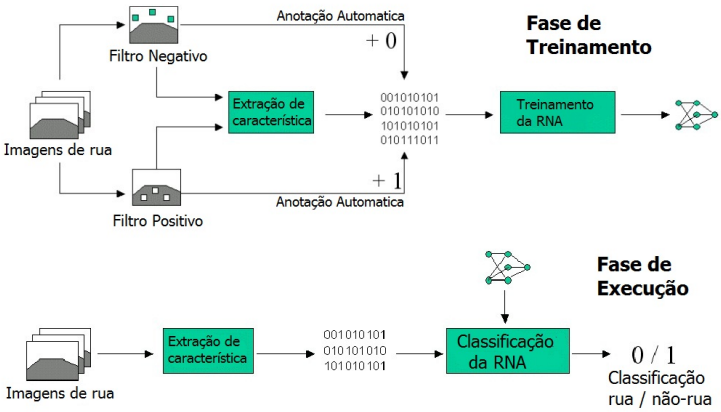
Nesse trabalho foi feita uma avaliação de identificadores baseados em cada característica isolada, ou seja, somente cor ou textura como mostra a Figura 2.4. Além disso, comparou essas avaliações com as avaliações da combinação dessas características, os resultados mostraram um ganho de desempenho. Ele também avaliou um sistema que possui um modelo de "rua" para cada imagem, ou um modelo para todas as imagens e finalmente um modelo para cada tipo de rua.

2.4 Modelo Adaptativo Utilizando RNA

O sistema de (Foedisch, 2004) utilizou redes neurais artificiais para realizar a tarefa de identificação de superfície navegável. O que significa uma etapa de treinamento com exemplos de padrões já classificados e uma etapa posterior de classificação de novos padrões, como mostra a Figura 2.5(b).



(a) Imagem onde os quadrados vermelhos representam os pixeis que são pré-classificados como não navegáveis enquanto os quadrados verdes englobam os pixeis navegáveis



(b) Diagrama de blocos do sistema na etapa de treinamento e execução. Imagem adaptada de (Foedisch, 2004).

Figura 2.5: Sistema desenvolvido por (Foedisch, 2004). A imagem (a) mostra um exemplo da forma de geração de padrões de treinamento e a imagem (b) mostra o diagrama do sistema na etapa de treinamento e execução.

Para essa abordagem ser possível, algumas regiões da imagem foram previamente definidas como navegáveis e não navegáveis - Figura 2.5(a). Além disso, os valores de entrada das RNA são a representação binária do histograma de cores RGB de sub-imagens de 7×7 pixeis assim como fez (Rasmussen, 2002). Como cada canal de cor pode ser representado por oito bits,

então o histograma é enviado para a RNA por meio de 24 neurônios de entrada. Além do histograma, a RNA recebe como entrada, o valor da posição x e y de cada sub-imagem normalizado totalizando 26 atributos de entrada.

2.5 Sistema Visual Para Controle de Velocidade

O processamento de imagens ajudou a equipe de Stanford vencer o *"DARPA Grand Challenge"* (DARPA, 2009). Como o sistema baseado no laser era capaz de mapear apenas 22 metros a frente do robô, a velocidade segura para que ele desviasse de possíveis obstáculos era de 25mph (aproximadamente 40km/h). Então foi utilizado processamento de imagens para classificar uma região de 70 metros à frente do robô para que, quando possível, sua velocidade aumentasse para 35mph (aproximadamente 56km/h).

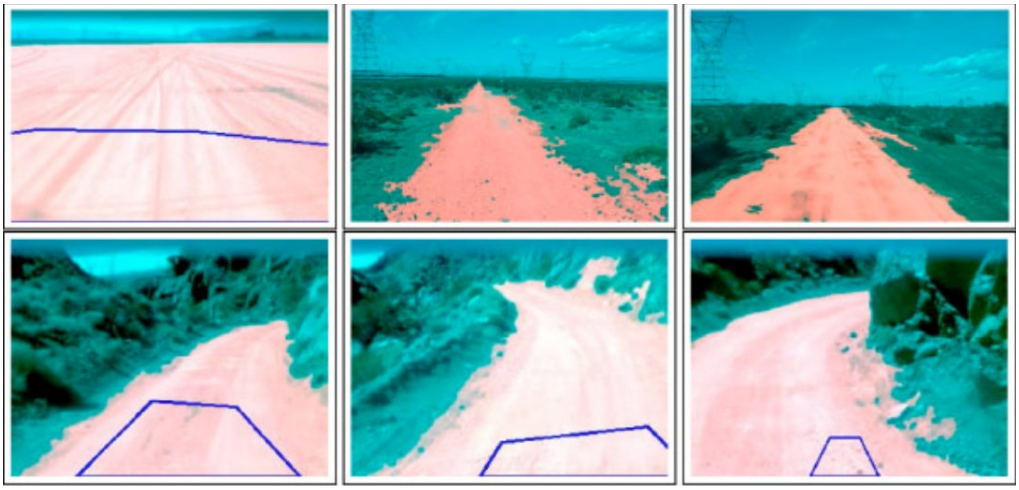


Figura 2.6: Execução do algoritmo de processamento de imagem desenvolvido por (Thrun *et al.*, 2006). Baseado no padrões encontrados na região considerada navegável pelo laser (quadrilátero azul), o sistema identifica a região navegável na imagem.

O método desenvolvido criou um quadrilátero à frente do robô definido como região navegável a partir dos dados obtidos do laser. Esse quadrilátero era projetado de maneira justa na imagem obtida da câmera e os pixeis dentro dele eram usados para treinar o conceito de região navegável no algoritmo de visão computacional.

O algoritmo de aprendizado mantinha uma mistura de gaussianas do modelo de cor do terreno navegável. Esse aprendizado possui dois possíveis caminhos; um ajuste lento para quando as condições de luz variam; e um ajuste rápido quando a região navegável da nova imagem possui outra cor (por exemplo, quando o robô se move de uma rua pavimentada para uma não pavimentada). A Figura 2.7 ilustra o segundo caso.

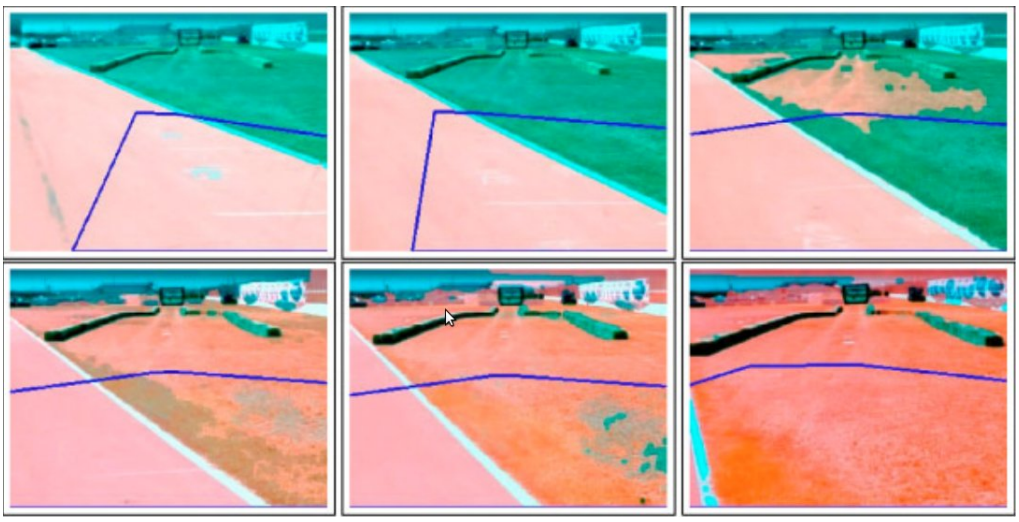


Figura 2.7: Rápida adaptação de Stanley. Utiliza o laser para definir os padrões de cor, faz com que o sistema se adapte mesmo quando a cor da superfície muda drasticamente.

O mapa visual criado pelo processamento de imagem não foi utilizado para o controle de direção, mas foi utilizado apenas para o controle de velocidade. Quando era detectado um corredor não navegável a uma distância de 40 metros, o veículo diminuía sua velocidade para 25mph, velocidade suficiente para o algoritmo baseado em informações do laser realizar uma navegação segura.

2.6 Considerações

Os trabalhos apresentados resolveram a identificação de região navegável de formas diferentes e limitadas. Thrun *et al.* (2006) utilizaram o processamento de imagens de forma secundária na navegação, enquanto Pomerleau (1996) controlava apenas a direção, além da necessidade de treinamento e diminuição da resolução da imagem. Tan *et al.* (2006) assumiu que uma região da imagem sempre será navegável, enquanto Rasmussen (2002) combinou informações da imagem com dados obtidos por um sensor laser para melhorar sua classificação. A distinção entre os trabalhos citados neste capítulo mostra a diversidade e a limitação do estado atual da arte. Todos esses trabalhos demonstram o quão complexo e extenso é a criação de um sistema de navegação baseado em visão.

Essa dissertação apresenta um sistema de identificação de superfícies navegáveis baseado em visão computacional e redes neurais artificiais similar a (Foedisch, 2004), entretanto não foi utilizado apenas as cores RGB de uma imagem, foram também utilizados vários descritores estatísticos, além de outros espaços de cores. O Capítulo 3 descreve detalhadamente o identificador proposto.

CAPÍTULO

3

Metodologia

O sistema de navegação autônoma baseado em visão pode ter seu funcionamento descrito da seguinte forma: uma câmera de vídeo, conectada a um robô, captura uma imagem de uma rua caso o robô utilizado seja um veículo. Para cada imagem, o sistema gera os atributos de todos os elementos da imagem. Cada um desses elementos, que podem ser um único pixel ou grupos de pixeis, são classificados de forma independente segundo a sua navegabilidade. Essa classificação utiliza as características ou atributos calculados na etapa anterior. A composição dessas classificações cria uma mapa visual de navegabilidade, que um algoritmo de navegação pode utilizar para determinar a ação a ser tomada, ações como *"seguir em frente"*, *"fazer uma curva"*, ou até mesmo *"parar"*. Após a execução dessa ação, o sistema observa novamente o ambiente, ou seja, a câmera captura uma nova imagem e o algoritmo recomeça. O diagrama da Figura 3.1 mostra cada uma dessas etapas detalhando os dados necessários para cada fase e os dados gerados para a próxima fase.

Este capítulo apresenta o detalhamento de cada etapa descrita no parágrafo anterior. A Seção 3.1 mostra como uma imagem é transformada em um conjunto de elementos que representam regiões da imagem. A Seção 3.2 descreve como funciona a etapa de classificação dos elementos. Nela também é esclarecido, de forma sucinta, o método de aprendizagem de máquina conhecido por *Redes Neurais Artificiais* (RNA) do tipo *Multi-Layer Perceptron* (MLP), uma vez que o classificador se utiliza desses métodos para determinar se um elemento da ima-

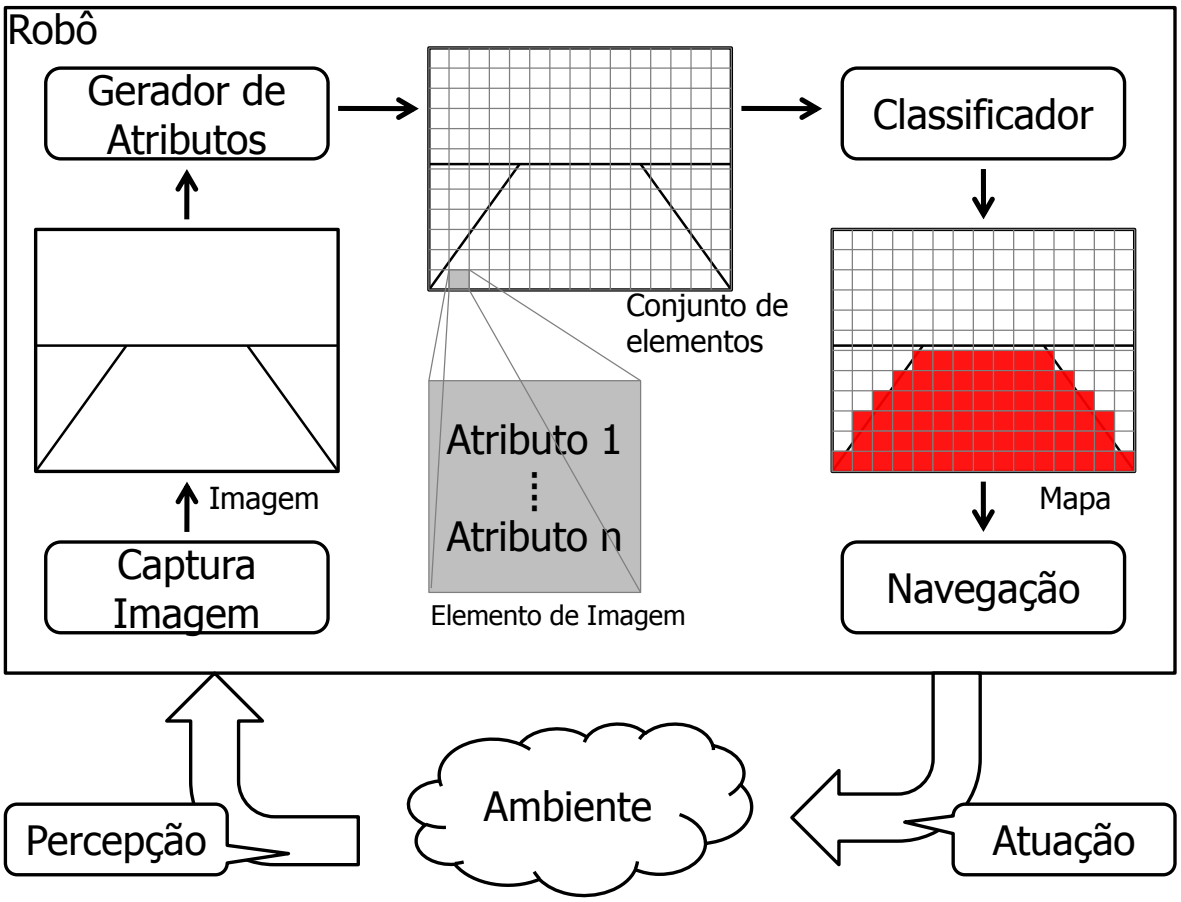


Figura 3.1: Diagrama do Sistema: mostra cada uma das etapas detalhando os dados necessários para cada fase. Os blocos pintados de vermelho no mapa são os elementos considerados navegáveis.

gem é navegável ou não. Por fim, a Seção 3.3 comenta um algoritmo de navegação reativa para mostrar como o "Sistema de Identificação de Superfícies Navegáveis" pode ser utilizado.

3.1 Gerador de Atributos

O gerador de atributos é responsável por transformar uma imagem - matriz de pixeis - em uma outra estrutura de dados, uma matriz de elementos. Portanto a menor unidade a ser classificada consiste em um bloco de pixeis. Essa transformação tem como propósito o aumento de desempenho e a avaliação de uma informação obtida a partir de um grupo de pixeis diretamente conectados.

Na área de visão computacional existem diversos algoritmos específicos e com bom desempenho para seus propósitos. Entretanto, na maioria das vezes, tais algoritmos não precisam ser

executados em tempo real. Como por exemplo no processamento de imagens médicas. Para um sistema de navegação autônomo, processamento em tempo real é essencial, o que justifica a necessidade de um bom desempenho.

Em relação à avaliação da informação de um grupo de pixeis, existem diversos exemplos na literatura que mostram bons resultados. (Lee *et al.*, 2005) utilizaram blocos de (9×9) pixeis. Para cada bloco foi gerada a média μ dos valores RGB e utilizou-se a fórmula de distância de Mahalanobis[1] $(x - \mu_1)^T \sum_{1}^{-1}(x - \mu_1)$, onde x e μ_1 são vetores na forma $[R, G, B]^T$ (Thorpe *et al.*, 1988), para verificar se o bloco era navegável ou não. (Foedisch e Takeuchi, 2004) apresentaram um método de *Road Detection* adaptativo, o qual usa um histograma RGB computado de uma sub-imagem de (7×7) pixeis para classificar um conjunto de pixeis como navegável ou não. Diferente desses e de outros trabalhos já existentes na literatura, neste trabalho vários outros atributos de blocos de pixeis foram gerados e combinados visando uma classificação mais robusta. Exemplos desses atributos são a entropia de um pixel e o uso de outros espaços de cores.

A Seção 3.1.1 descreve com mais detalhes a transformação da imagem para a matriz de elementos. A Seção 3.1.2 revisa de forma sucinta vários conceitos de processamento de imagens e descreve a forma de calcular cada atributo utilizado neste trabalho. E por último, a Seção 3.1.3 discute o resultado do gerador de atributos antes dessa dissertação iniciar a discussão sobre a classificação destes elementos.

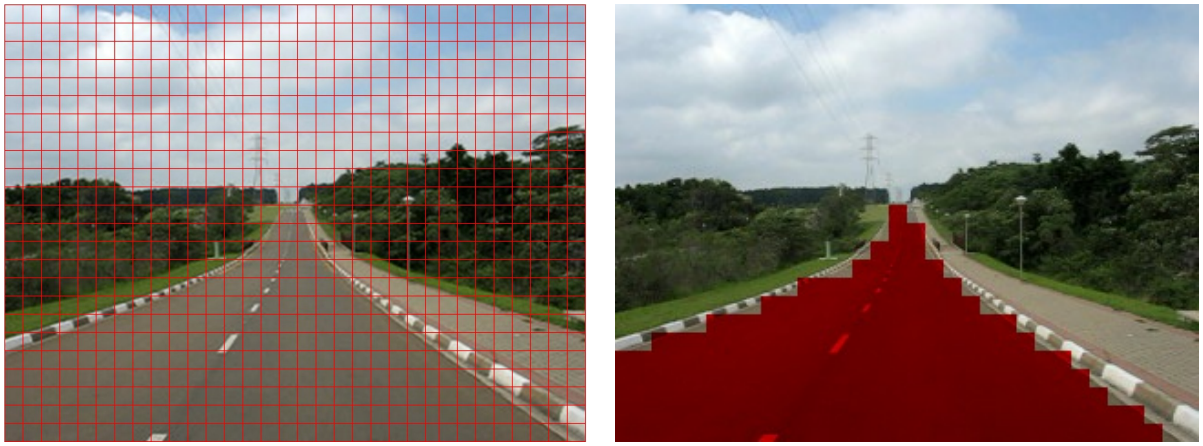
3.1.1 Método de Segmentação Baseada em Blocos

Este método consiste em tratar e avaliar um conjunto de pixeis diretamente conectados, ou seja vizinhos, como um grupo (bloco). A partir deste grupo é gerado um único atributo ou característica para representá-lo. Esta característica pode ser a média dos valores RGB, entropia dos pixeis e muitas outras medidas explicadas na Seção 3.1.2.

Este método propõe dividir a imagem em grupos de dimensão $(K \times K)$, como mostra a Figura 3.2(a), onde cada quadrado é um grupo. Supondo a matriz I sendo uma imagem de resolução igual a $M \times N$. O valor do elemento $I(m, n)$ corresponde ao valor do pixel na m-ésima linha e n-ésima coluna da imagem, onde $(0 \leq m < M)$ e $(0 \leq n < N)$. Assim, o grupo $G(i, j)$ contém todos os pixeis $I(m, n)$, tal que $(\ (i * K) \leq m < ((i * K) + K)\ )$ e $(\ (j * K) \leq n < ((j * K) + K)\ )$.

[1]A distância de Mahalanobis é baseada nas correlações entre variáveis. É uma métrica comum que tenta captar as propriedades não isotrópicas de um espaço de características d-dimensional. Seu cálculo varia de acordo com a variação estatística de cada componente usando uma matriz de covariância (Grudic e Mulligan, 2006).

Assim, se o valor da característica que representa o grupo $G(i, j)$ for considerado como um valor que representa uma região navegável na imagem, então todos os pixeis pertencentes a G serão considerados navegáveis. A Figura 3.2(b) mostra os pixeis dos grupos considerados navegáveis pintados de vermelho.



(a) Imagem de (320×240) pixeis fatiada em pequenos quadrados de dimensão (10×10) pixeis

(b) Exemplo de uma imagem classificada: Os pixeis dos grupos considerados navegáveis estão pintados de vermelho

Figura 3.2: Segmentação de imagem baseada em blocos.

Repassar a mesma classificação do valor da característica do grupo para todos os seus pixeis faz com que seja reduzida a precisão nas bordas. Entretanto, essa imprecisão, em geral, é aceitável uma vez que o ganho de desempenho é substancial dependendo de K, M e N. Por exemplo, ao considerar apenas um atributo, utilizando imagens de dimensão (320×240) e dividindo-as em blocos de dimensão (10×10), obtêm-se apenas 768 valores em vez de 76.800 para serem computados por um método de classificação.

Conforme discutido na Subseção 3.1.2.2, o valor gerado para cada bloco de pixeis pode também ser interpretado como a textura da região referente ao bloco. Outra vantagem da segmentação baseada em blocos é a possibilidade de comparação de histogramas ou qualquer medida estatística de primeira ordem dos blocos. Pois valores gerados a partir de um bloco de pixeis podem facilmente adicionar o contexto de localização dentro da imagem utilizando a localização do bloco.

3.1.2 Atributos de Imagem

Como descrito anteriormente, este trabalho utiliza vários atributos de imagem para determinar se um bloco de pixeis representa ou não, parte de uma região navegável. Tais atributos são

medidas estatísticas calculadas sobre o conjunto de pixeis de cada bloco. Esta seção revisa vários conceitos de processamento de imagens e também descreve a forma de calcular cada medida estatística utilizada.

Neste trabalho foram utilizados apenas cor e textura, portanto essa dissertação se restringe a descrever de forma sucinta apenas estes dois itens, além de alguns conceitos necessários para o entendimento do cálculo dos atributos.

3.1.2.1 Cor

Um espaço de cor é uma representação matemática para descrever cores através de tuplas de números. Dentre os espaços de cores conhecidos destacam-se: RGB, HSV, YUV, entre outros. A seguir, uma breve descrição de alguns deles.

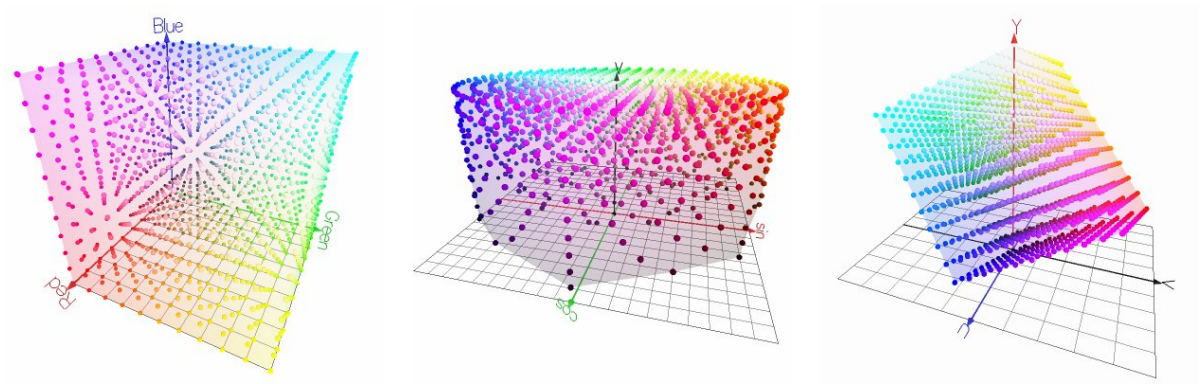
- O espaço de cores RGB é um espaço onde cada pixel pode ser definido pela combinação de intensidades das cores vermelho (*red*), verde (*green*) e azul (*blue*) (Joblove e Greenberg, 1978). A Figura 3.3(a) ilustra o espaço RGB.
- O espaço de cores HSV é um espaço onde cada pixel pode ser definido pelas componentes matiz (*hue*), saturação (*saturation*) e brilho (*value*) (Reiter, 2004). A representação gráfica do HSV é um cone ou um cilindro, como mostra a Figura 3.3(b).
- O espaço de cores YUV, Figura 3.3(c), é similar com o espaço YCbCr usado em televisores com sistema PAL. A componente Y é a componente de brilho. As componentes U e V descrevem a tonalidade e a saturação da cor, sendo a componente U responsável pela tonalidade azul e a V pela tonalidade vermelha (Reiter, 2004).

3.1.2.2 Textura

Não existe uma definição formal única para o conceito de textura. Na literatura pode-se encontrar várias abordagens e definições, como por exemplo a definição de L. V. Gool (Gool *et al.*, 1985) que a descreveu como:

> *"Uma estrutura composta de uma área contendo um número grande de elementos relativamente ordenados ou padrões, que individualmente não representam muito. Mas se observados globalmente, tais elementos podem ser usados para identificar uma determinada área."*

ou a definição de R. M. Haralick (Haralick, 1979) que disse:



(a) RGB (b) HSV (c) YUV

Figura 3.3: Representações gráficas dos espaços de cores utilizados. Imagens retiradas de (Couleur, 2010).

> *"Uma imagem com textura pode ser descrita pelo número e tipos de suas primitivas, pela organização espacial ou layout de suas primitivas. A organização espacial pode ser aleatória ou pode obedecer um padrão relativamente uniforme entre uma primitiva e suas primitivas vizinhas. Esta dependência pode ser estrutural, probabilística ou funcional."*

Desta forma, pode-se definir textura como uma propriedade ou uma característica de um conjunto de pixeis adjacentes.

3.1.2.3 Histograma

Na área de estatística, um histograma é uma representação gráfica de uma distribuição de frequências de um conjunto de dados. Em processamento de imagens pode-se definir um histograma como um mapeamento que demonstra um valor numérico $h(n)$ para cada nível de cinza n de uma imagem (para imagens em escala de cinza). O valor $h(n)$ corresponde ao número de vezes que cada nível de cinza n ocorre na imagem (Costa e Cesar, 2000). Medidas baseadas nesse tipo de histograma são consideradas medidas estatísticas de primeira ordem. Em imagens coloridas cria-se um histograma para cada canal do espaço de cores da imagem, se ela é representada por RGB então cria-se um histograma para o vermelho, outro para o verde e outro para o azul.

Seja I uma imagem de dimensão $(M \times N)$ que possui um número de cores C. Um histograma pode ser matematicamente definido da seguinte forma (Roncatti, 2008):

$$h(i) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \delta(i, I(m,n)), \qquad i = 0, 1, 2, ..., (C-1), \tag{3.1}$$

$$\delta(i,j) = \begin{cases} 1 & \text{se} \quad i = j \\ 0 & \text{se} \quad i \neq j \end{cases}.$$

A função $\delta(i,j)$ é chamada delta de Kronecker.

3.1.2.4 Descritores Estatísticos

Descritores estatísticos são propriedades que representam a probabilidade de se encontrar um pixel, análise de distribuição ou regularidade na imagem. Nesta seção serão descritas algumas medidas utilizadas neste trabalho.

Dado um histograma ($h(i) \quad | \quad i \in C$) de uma imagem de dimensão $(M \times N)$ é possível gerar várias medidas para caracterizar essa imagem. Onde C é o número de cores da imagem.

$$p(i) = h(i)/(M * N).$$

- Média:

$$\mu = \sum_{i=0}^{C-1} i * p(i). \tag{3.2}$$

- Variância:

$$\sigma^2 = \sum_{i=0}^{C-1} (i - \mu)^2 * p(i). \tag{3.3}$$

- Energia:

$$\varepsilon = \sum_{i=0}^{C-1} (p(i))^2. \tag{3.4}$$

- Entropia:

$$E = -\sum_{i=0}^{C-1} p(i) * log_2(p(i)). \tag{3.5}$$

A entropia de Shannon pode ser definida, de maneira simplificada, como grau de regularidade de um conjunto de dados (Shannon, 1948).

3.1.3 Considerações

Além dos atributos apresentados, existem inúmeros outros que podem ser calculados a partir de uma imagem. Tais atributos podem depender de apenas um pixel, de um conjunto de pixeis, ou até mesmo da imagem inteira. Este trabalho combina um conjunto de atributos de elementos

de imagem para realizar a classificação. Isso significa que para cada elemento de imagem foram calculadas diversas medidas estatísticas de cada canal de cor para cada espaço de cor. Ou seja, foram calculados atributos como média de cada canal RGB, HSV e YUV, assim como a entropia, variância e energia, para cada bloco de pixel gerando os elementos de imagem. O Apêndice B traz uma lista de todos os atributos calculados para cada elemento.

Uma vez que tem-se muitos atributos e alguns deles são redundantes, é necessário escolher quais deles ou quais combinações utilizar para identificar a região navegável em ambientes externos visando acurácia na identificação e um desempenho em tempo real de todo o sistema. Para realizar esta tarefa foi utilizado o seletor de atributos baseado em medidas de saliências de uma RNA proposto por (dos Santos, 2007). A Seção 4.1 descreve o funcionamento desse seletor e um experimento realizado para verificar a taxa de acerto de diferentes combinações de atributos de imagem e para selecionar os que admitem bons desempenhos na classificação.

Uma vez descrita a forma de geração e seleção dos elementos de imagem e descrita a forma de calcular seus vários atributos, fica nítido como a imagem é processada antes de iniciar a etapa de classificação. Estes elementos selecionados compõem uma matriz que ao invés de possuir apenas as informações das cores dos pixeis, também possui valores que representam por exemplo o grau de regularidade ou a distribuição dos valores de cada canal de cor dentro de um bloco de pixel. Essas medidas são utilizadas pelo classificador para este determinar se este elemento de imagem representa uma região navegável ou não navegável.

3.2 Classificador

O classificador é responsável por determinar se um elemento de imagem representa uma parte da imagem que mostra uma região do ambiente navegável ou não navegável. Para determinar a qual classe um bloco de pixeis pertence são utilizados os atributos calculados pelo gerador de atributos. Um classificador pode ser definido da seguinte maneira:

Seja $\{x\}$ um elemento de imagem e $C = \{C_1, \ldots, C_n\}$ o conjunto de todas classes possíveis para uma determinada classificação. O classificador recebe todos ou alguns atributos de x como entrada e retorna a qual classe esse elemento pertence. A Figura 3.4 retrata esse processo.

Para executar a tarefa de classificação foi utilizado um método de aprendizado de máquina conhecido por *"Redes Neurais Artificiais"* - do tipo *"Multi-Layer Perceptron"* (MLP). A Seção 3.2.1 faz uma breve revisão dos conceitos de aprendizado de máquina mais especificamente sobre RNA. A Seção 3.2.2 comenta alguns métodos de avaliação de RNA, descreve o método utilizado neste trabalho e explica o porque do seu uso. A Seção 3.2.3 descreve de forma geral sobre as etapas de treinamento e uso do classificador, bem como toda sua estrutura.

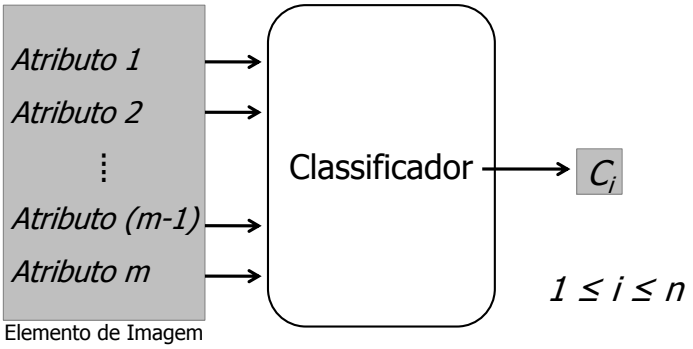


Figura 3.4: Diagrama que descreve o funcionamento de um classificador. O classificador utiliza os valores dos atributos de um elemento de imagem para determinar qual classe ele pertence.

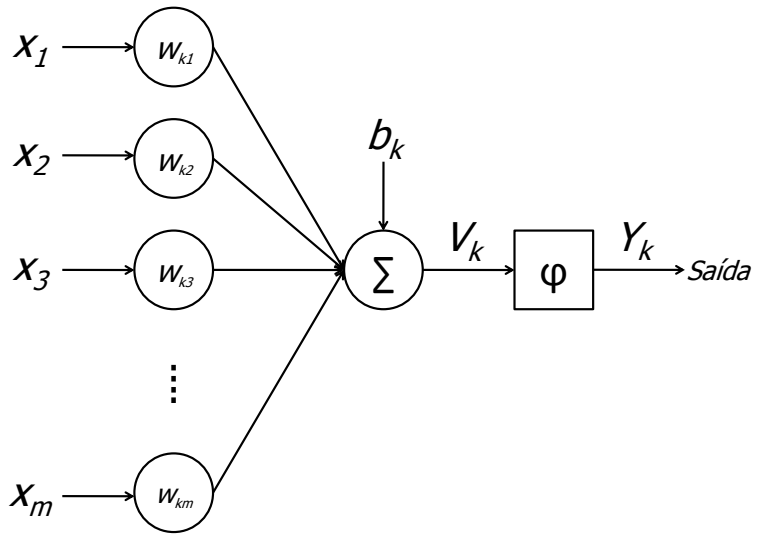
3.2.1 Aprendizado de Máquina

O aprendizado de máquina é uma área da Inteligência Artificial (IA) que estuda métodos computacionais que automatizam a aquisição de conhecimento (Baranauskas e Monard, 2003). O aprendizado de máquina divide os métodos de aprendizagem em três categorias: aprendizado supervisionado, aprendizado não-supervisionado e aprendizado por reforço (Baranauskas e Monard, 2003) (Mitchell, 1997) (Carbonell *et al.*, 1984).

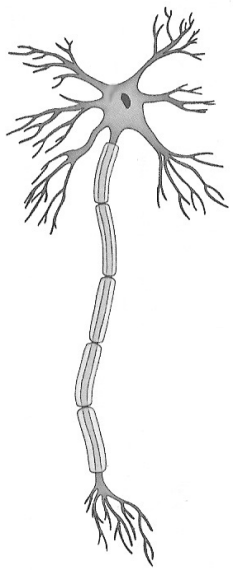
Um método de aprendizado supervisionado pode ser descrito como: um modelo que recebe um conjunto de pares de valores (entradas e saídas) rotulados que representam um determinado ambiente. Para que o modelo represente o ambiente, é necessário que ele passe por uma fase de treinamento. Nesta fase, o modelo recebe os valores que representam o ambiente (entradas) e para cada um deles é verificado se foi gerada a saída correspondente. Caso isto não ocorra, o modelo é reestruturado de forma a produzir o resultado correto (Mitchell, 1997). Exemplos de métodos desta categoria são RNA (MLP), *support vector machines*, kNN (*k-Nearest Neighbour*) e árvores de decisão.

As RNA buscam a solução dos problemas por meio da simulação computacional de mecanismos e estruturas do cérebro humano. De forma análoga ao modelo biológico, as RNA são formadas por um conjunto de neurônios e conexões que são responsáveis pelo processamento de informações. O neurônio artificial, ilustrado na Figura 3.5(a), é composto de três elementos básicos: conexões entre neurônios que permitem propagar dados através da rede; um somador que efetua a somatória dos dados que chegam ao neurônio; e uma função de ativação que determina o valor do dado de saída de acordo com o resultado do somador. Portanto uma RNA

consiste basicamente em um conjunto de neurônios artificiais conectados e um algoritmo de aprendizado.



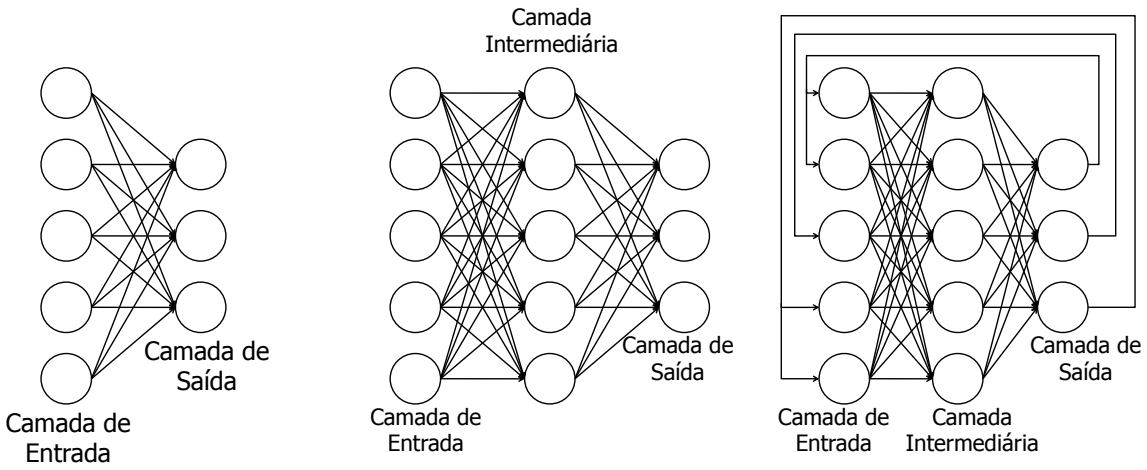
(a) As entradas são denotadas por $\{x_1, \ldots, x_m\}$, sendo $\{w_{k1}, \ldots, w_{km}\}$ os seus respectivos pesos. O símbolo $\sum$ representa o somador da rede, b_k é o bias do neurônio que tem a finalidade de aumentar ou diminuir o valor de entrada para a função de ativação, V_k é o resultado do somador, φ representa a função de ativação e Y_k é o dado de saída do neurônio (Haykin, 1998)



(b) Neurônio Biológico

Figura 3.5: Estrutura básica de um neurônio.

A forma como os neurônios estão conectados, também chamada topologia, influencia diretamente no aprendizado da RNA. Tradicionalmente, as RNA possuem uma topologia onde seus neurônios ficam dispostos em camadas. Essas topologias são classificadas em três grupos principais: *perceptron*, MLP e recorrente. O ***Perceptron*** (Figura 3.6(a)) ou também conhecido como *single-layer feedforward* consiste na configuração mais básica de RNA. Composta de apenas uma camada de entrada e outra de saída e sem ciclos entre conexões dos neurônios, esta RNA fica limitada à resolução de problemas lineares (Braga *et al.*, 2000). O ***Multi-Layer Perceptron*** (Figura 3.6(b)) ou *multi-layer feedforward* adiciona as camadas ocultas ao modelo *perceptron* tradicional elevando a capacidade de resolução para problemas não lineares, bem como melhoria da precisão dos resultados da RNA (Haykin, 1998). Na topologia ***Recorrente*** (Figura 3.6(c)) há a presença de um laço que conecta a saída do neurônio para a entrada de um outro, de forma a realimentar a RNA. As redes recorrentes possuem a capacidade de simular uma memória de curto prazo. Estruturas recorrentes admitem também um número indefinido de camadas ocultas (Rojas, 1996).



(a) Perceptron (b) Multi-Layer Perceptron (c) Recorrente

Figura 3.6: Representações gráficas das topologias de RNA.

O algoritmo de aprendizado escolhido é importante tanto para a corretude da resposta da RNA quanto para seu desempenho no tempo de treinamento. O algoritmo de aprendizado deve ser executado antes da utilização da RNA na resolução de um problema. Durante essa execução, os pesos das conexões são ajustados para a melhor correspondência entre entradas e saídas dos exemplos fornecidos para a fase de treinamento (Russell e Norvig, 2003). Alguns dos algoritmos de aprendizado mais utilizados são o algoritmo *backpropagation* (Rumelhart *et al.*, 1986) e suas derivações como *resilient backpropagation* (Riedmiller e Braun, 1994).

3.2.2 Métodos de Avaliação de Desempenho do Classificador

Como o classificador utilizado neste trabalho é composto de RNA, é necessário dar maior importância aos métodos de avaliação, ou seja, aos métodos que determinam quando a RNA está pronta para ser utilizada. Existem inúmeras formas de se avaliar uma RNA, nesta seção serão discutidas duas formas bem fundamentadas na literatura e será apresentado uma terceira que foi utilizada neste trabalho.

Um dos métodos mais utilizados é treinar uma RNA até que seu "erro quadrático médio" (EQM) seja igual a zero ou que chegue a um valor mínimo aceitável - sua função pode ser definida matematicamente como a Equação 3.6.

$$EQM = \frac{1}{2N} \sum_{i=1}^{N} ||d_j - F(x_j)||^2. \tag{3.6}$$

Na Equação 3.6, N é o número de padrões utilizados na RNA, d_j é o resultado esperado e $F(x_j)$ é o resultado obtido pela RNA. Entretanto, vale ressaltar que um EQM pequeno nem sempre implica em uma boa generalização (Haykin, 1998). Além disso, o EQM não nos informa quantos padrões foram mal classificados, ou seja, se o erro é alto em apenas alguns padrões ou se o erro está uniformemente espalhado entre os resultados. Suponha um conjunto de dez padrões apresentados a uma RNA, com $EQM = 0,2$, esse valor pode significar que os dez padrões possuem um erro de $0,2$ ou então, que três padrões possuem erro $= 0,05$, dois com erro de $0,1$ e cinco com erro igual a $0,33$. A falta dessa informação faz com que em alguns casos, seja escolhido outro meio de avaliação onde se observa a quantidade de respostas corretas.

Outra forma de avaliar o desempenho de uma RNA é verificar quantos padrões obtiveram uma classificação correta, atribuindo uma porcentagem de acerto para a RNA (*HIT RATE*). Para computar esse número é necessário definir uma precisão, pois nem sempre o valor de saída da RNA é exatamente o valor esperado. Por exemplo, uma RNA é treinada para retornar 0 ou 1 em um de seus neurônios da camada de saída. Entretanto, para alguns padrões ela retorna um valor de $0,85$ em vez de 1, se a precisão for de $0,1$ então esses padrões foram classificados erroneamente; se a precisão for $0,2$ então a classificação está correta. Portanto, utilizar esse método implica em conhecer muito bem o conjunto de dados utilizado para definir uma boa precisão. Uma precisão alta demais pode considerar muitos resultados como erro, fazendo a taxa de acerto da RNA ficar muito baixa, e uma precisão baixa demais pode gerar uma taxa de acerto elevada, mas um resultado final insatisfatório.

Visando uma classificação mais adequada foi utilizado um método que atribui uma nota a RNA utilizando o erro de cada padrão assim como o EQM junto com a informação de precisão usada no *HIT RATE* sobre esse resultado. A seguir, a descrição deste método para uma RNA com 1 neurônio de saída treinada para retornar 0 ou 1, utilizando uma precisão de $0,1$.

Primeiramente, foram discretizados os possíveis resultados de acordo com a precisão desejada, para esse caso de exemplo tem-se 11 valores: $\{0,0;\ 0,1;\ 0,2;\ ...;\ 0,8;\ 0,9;\ 1,0\}$. Nosso objetivo agora é criar um histograma desses valores, onde estes significam o erro mínimo obtido para cada padrão. Em outras palavras, tem-se um contador de frequência para o erro no valor $0,0$ a $0,09$, outro contador para $0,1$ a $0,19$ e assim por diante.

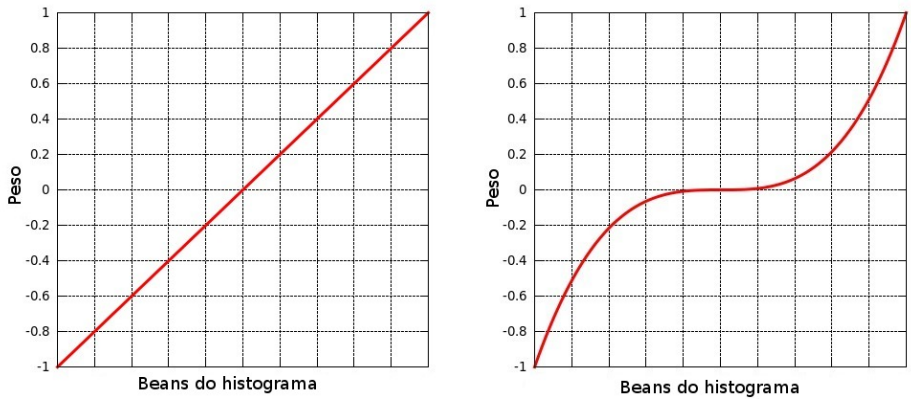
Após inicializar os contadores, para cada padrão apresentado a RNA, é calculado o erro $(d_j - F(x_j))$. Esse erro é posteriormente truncado na primeira casa decimal para associá-lo aos valores discretizados do histograma, ou seja, se o erro for $0,23$ então o contador de "$0,2$" é incrementado, se o erro for $0,09$ então o contador de "$0,0$" é incrementado e assim por diante.

Após analisar todos os padrões, obtêm-se a frequência de cada nível de erro discretizado, ou seja, seu histograma. Pode-se então utilizar a taxa de acerto para cada nível de erro separadamente. Entretanto, para este trabalho foi decidido unir essas informações aplicando um peso

para cada nível, somando seus resultados e por fim normalizando o resultado final gerando uma nota, neste trabalho essa nota foi definida como (FIT). Como mostra a Equação 3.7.

$$FIT = \frac{\frac{1}{N*p(0)}\left(\sum_{i=0}^{hmax} h(i) * p(i)\right) + 1}{2.0}, \tag{3.7}$$

onde $h()$ é o histograma gerado, $h(i)$ representa o erro $0, i$ e $h(10)$ representa o erro máximo $1, 0$. O peso utilizado por cada nível i de erro é representado por $p(i)$. N é o número de padrões apresentados a RNA, e $hmax$ é o número de *beans* do histograma.



(a) Exemplo de função linear (b) Exemplo de função exponencial

Figura 3.7: Funções aplicadas como pesos nos valores do histograma.

Os pesos $p(i)$ podem ser ajustados de acordo com a função utilizada. A Figura 3.7 mostra duas funções que podem ser utilizadas para gerar os pesos. A escolha de uma função depende do quão importante é cada resposta parcialmente errada. A função linear faz com que todos os valores sejam próximos, portanto não existe uma frequência do histograma que vale muito mais que a sua vizinha. Já em uma função exponencial, os extremos pesam muito mais que o centro do histograma. A Tabela 3.1 mostra os pesos resultantes destas duas funções.

Tabela 3.1: Pesos utilizados para calcular o FIT. Cada peso é multiplicado por um determinado *bean* do histograma.

Função	Erros discretizados										
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
Linear $(x/5)$	1.0	0,8	0,6	0,4	0,2	0,0	-0,2	-0,4	-0,6	-0,8	-1,0
Exponencial $(x/5)^3$	1.0	0,512	0,216	0,064	0,008	0,0	-0,008	-0,064	-0,216	-0,512	-1,0

Utilizar esse método híbrido que calcula o FIT, faz com que se obtenha uma métrica de avaliação parecida com o EQM, onde se atribui maior importância para os padrões classificados

corretamente, mas ainda considera-se os classificados erroneamente. Essa consideração pode afetar a nota tanto positivamente - quando muitos padrões chegaram perto de serem classificados corretamente mas não o são - quanto negativamente - quando os padrões são inversamente classificados. Devido a sua natureza, o FIT é mais sensível a ruídos que o EQM, ou seja, pode ter uma oscilação maior se comparado com o EQM, entretanto isso permite ao supervisor determinar com que tipo de dados está trabalhando e selecionar os melhores. A Seção 4.2 descreve um experimento realizado mostrando as similaridades e diferenças entre esse método que calcula FIT, e os métodos que calculam EQM e *HIT RATE*.

3.2.3 Modelo Final do Classificador e Considerações

Uma vez discutido o conceito de RNA e explicado sua forma de avaliação de desempenho pode-se entender como funciona o sistema de identificação proposto. Esta seção explica a arquitetura e funcionamento do modelo final do classificador.

Depois de escolher quais atributos de imagem serão utilizados na RNA - discutido na Seção 3.1 -, pode-se definir sua topologia. O número de atributos utilizados determina o número de neurônios da camada de entrada da RNA. Experimentos anteriores (Shinzato *et al.*, 2010), compararam 4 diferentes topologias de RNA diferenciadas apenas pelas camadas intermediárias (1 camada com 5 neurônios, 2 camadas com 5 neurônios cada, 1 camada com 10 neurônios e 2 camadas com 10 neurônios cada). Os resultados desse experimento mostraram não haver diferença significativa entre a RNA possuir 1 ou 2 camadas intermediárias, e mostraram que a RNA com 10 neurônios não apresentou melhoria significativa em relação a RNA com 5 neurônios na camada intermediária. Devido a isso, foi determinado que 5 neurônios na camada intermediária é suficiente para diferenciar regiões navegáveis das não-navegáveis nesta aplicação. Uma vez que quanto mais simples a RNA, maior a velocidade de treinamento e melhor pode ficar a generalização. Essa RNA é treinada para retornar o valor “1” quando o elemento avaliado representa uma região navegável e “0” quando o contrário, portanto a camada de saída possui apenas 1 neurônio. A Figura 3.8 mostra a topologia de RNA escolhida.

Assim como qualquer método de aprendizado supervisionado, nossa RNA deve passar por uma fase de treinamento. Na inicialização do sistema, o usuário seleciona pelo menos duas regiões disjuntas na imagem, onde uma é marcada como navegável e a outra como não navegável (Figura 3.9). A partir deste momento, o sistema captura algumas imagens para gerar os elementos de imagem e armazená-los para o treinamento da RNA. Para tentar conseguir um bom conjunto de dados de treinamento, a captura das imagens deve ser realizada em diferentes posições do cenário.

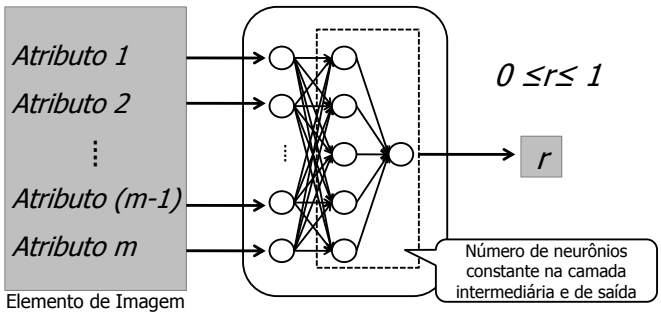


Figura 3.8: Topologia da RNA utilizada neste trabalho como classificador. Os atributos do elemento são as entradas da RNA que retorna um resultado pelo neurônio da camada de saída.

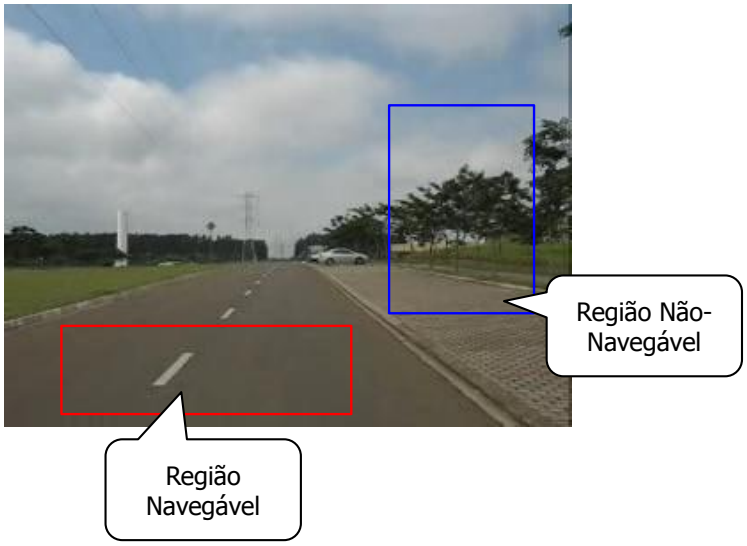


Figura 3.9: Inicialização do Sistema: o quadrado vermelho engloba os blocos de pixel considerados navegáveis pelo usuário, e o quadrado azul os considerados não navegáveis.

Após a aquisição das imagens e geração dos elementos, a RNA passa pelo processo de treinamento utilizando o algoritmo *"Resilient Propagation"*. Esse processo é realizado até 5 mil ciclos de treinamento ou até a RNA alcançar um FIT e *HIT RATE* aceitáveis - cada cenário pode alcançar um FIT diferente dependendo da dificuldade do cenário analisado. Quando a RNA chega a 5 mil ciclos de treinamento, ela é re-instanciada e treinada novamente sob as mesmas condições. Cabe ao usuário determinar o FIT e *HIT RATE* mínimos aceitáveis, observando os resultados obtidos. O *HIT RATE* é utilizado para garantir que um número mínimo de padrões seja classificado corretamente com a precisão mais alta possível. O experimento da Se-

ção 4.3 demonstra os desempenhos obtidos em diferentes condições de iluminação e diferentes cenários.

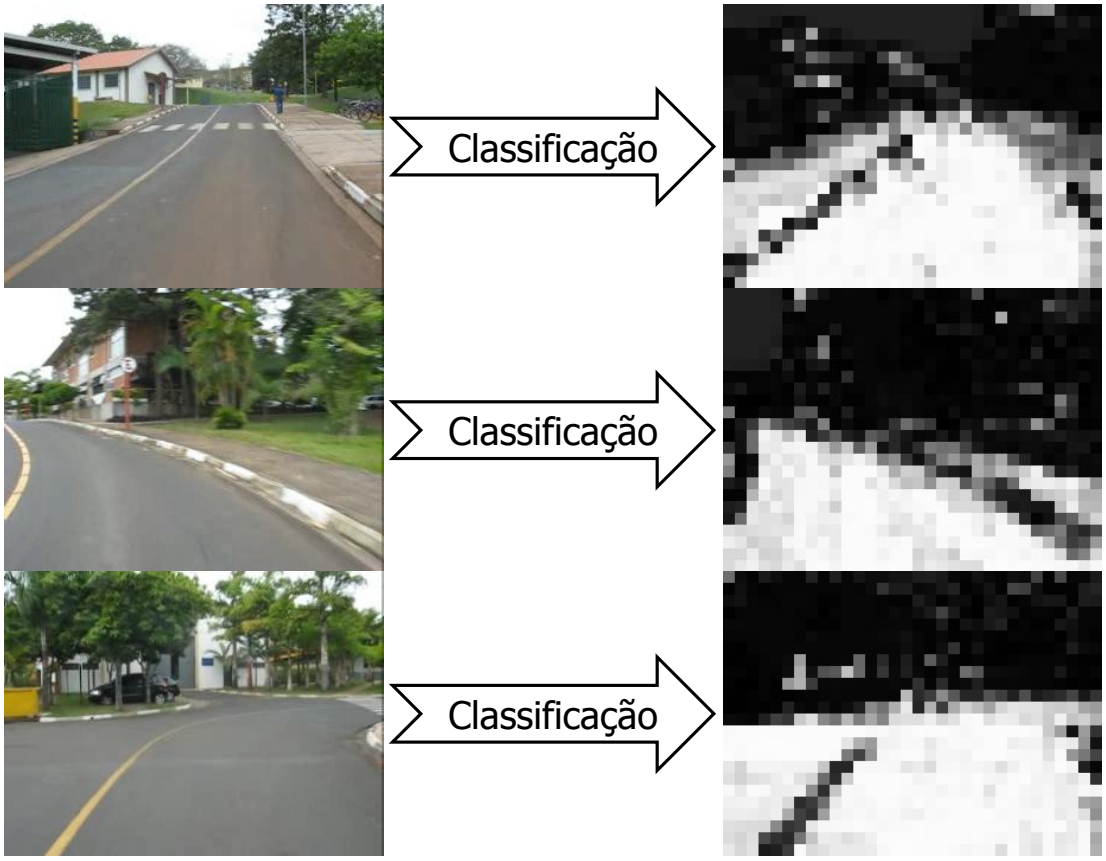


Figura 3.10: Resultados obtidos da classificação de algumas imagens. As imagens a esquerda são as imagens obtidas pela câmera. As imagens a direita mostram o resultado da classificação onde quanto mais branco maior a certeza de navegabilidade.

Após a execução da RNA para cada elemento de imagem, obtêm-se um mapa visual de navegabilidade como mostra a Figura 3.10, onde o resultado da classificação é convertido para uma imagem em tons de cinza. Quanto mais claro a cor do bloco, mais próximo de 1 foi o resultado da saída da RNA para o respectivo elemento. Desta forma, os blocos pintados de branco são as regiões consideradas navegáveis com a maior certeza possível e os blocos pintados de preto são as regiões não-navegáveis também com a maior certeza possível.

O diagrama apresentado na Figura 3.11 mostra como funciona nosso sistema depois da etapa de treinamento. É importante ressaltar que a entidade "Navegação" não faz parte do sistema de identificação e pode ser qualquer método capaz de tomar decisões a partir dos mapas de navegabilidade gerados. Esta poderia ser a forma final de um sistema de identificação de superfícies navegáveis. Entretanto, foi constatado que diferentes cenários e condições de iluminação pos-

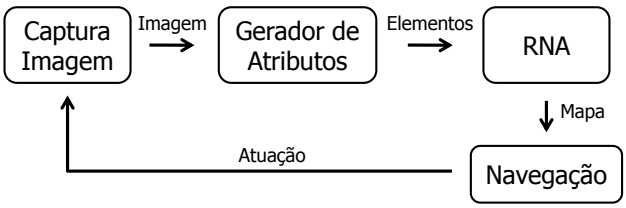


Figura 3.11: Diagrama do Sistema de Identificação Baseado em uma RNA.

suem diferentes atributos que podem alcançar as melhores classificações. Um exemplo é uma RNA que utiliza o atributo *média de B*, a qual obtém bons resultados sobre a diferenciação entre a rua asfaltada, estacionamentos e calçadas de concreto. Só que nesses casos, essas RNA erram mais que as outras em ambientes com ruas asfaltadas com sujeira (Shinzato e Wolf, 2010b). Devido a isso, foi criada uma nova abordagem publicada em (Shinzato e Wolf, 2010a) que consiste na identificação baseada na combinação de RNAs. Nessa nova abordagem, múltiplas RNAs classificam a imagem entre navegável e não-navegável. Para cada elemento de imagem, é calculada a média dos valores de saída das RNAs, essa média é o resultado final da classificação do elemento. A principal diferença entre as RNA é o conjunto de atributos utilizado como entrada para realizar a classificação. A Figura 3.12 mostra um diagrama detalhando essa nova abordagem.

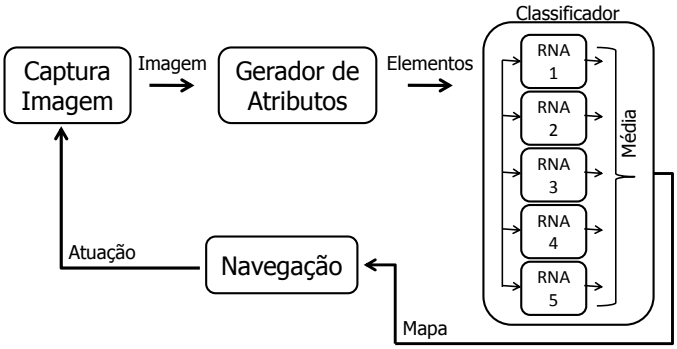


Figura 3.12: Diagrama do Sistema de Identificação Baseado em múltiplas RNAs.

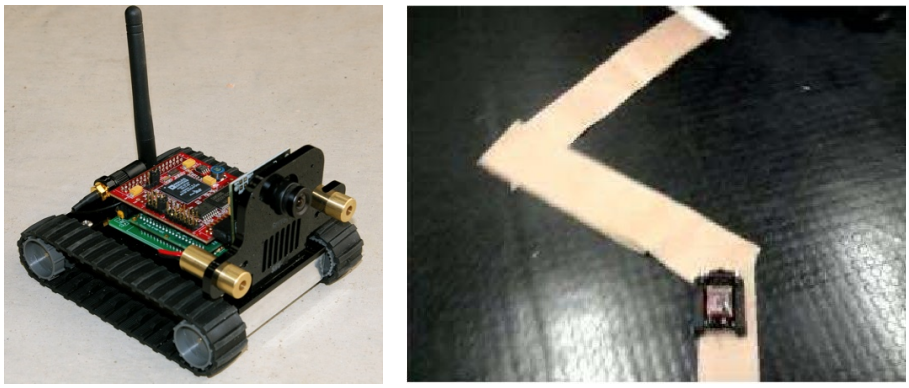
Como o número de neurônios da RNA é pequeno, seu tempo de treinamento também é reduzido. Assim é possível instanciar várias RNA para compor uma única classificação. Ex-

perimentos utilizando esse sistema de identificação foram realizados em diferentes plataformas (Sales *et al.*, 2010) e (Souza *et al.*, 2011) e apresentaram bons resultados.

3.3 Navegação

Uma vez definido como funciona nosso sistema de identificação, é necessário apenas definir um algoritmo de navegação baseado no mapa de navegabilidade gerado para que um robô possa navegar por um ambiente. Esta seção comenta, de forma sucinta, alguns experimentos realizados no Laboratório de Robótica Móvel (LRM[2]) do ICMC que utilizaram o método de identificação de superfícies navegáveis.

3.3.1 Navegação Autônoma Baseada em Visão Utilizando Controle Baseado em Autômato Finito Determinístico



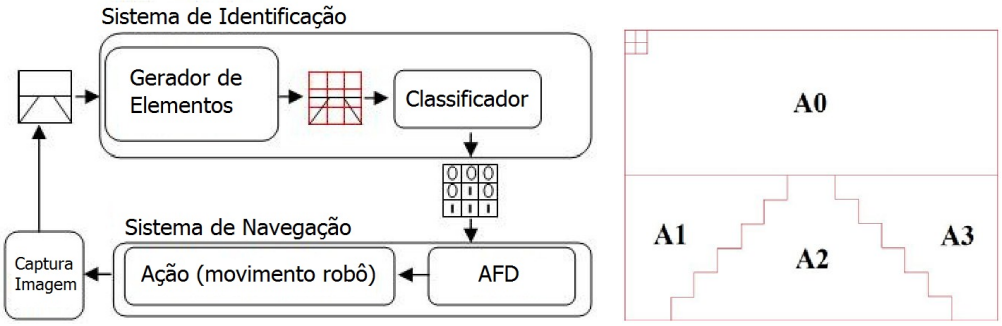
(a) SRV-1Q (b) Cenário utilizado

Figura 3.13: Experimento utilizando o Sistema de Identificação de Superfícies Navegáveis proposto. Imagens retiradas de (Sales *et al.*, 2010).

O trabalho de (Sales *et al.*, 2010) propõe o uso de autômatos finitos para realizar a tarefa de tomada de decisão para a navegação de um robô de pequeno porte, o SRV-1Q (Figura 3.13(a)), em um ambiente interno adaptado para sua navegação (Figura 3.13(b)). Utilizando o sistema de identificação de superfícies navegáveis e um método de avaliação sobre o mapa de navegabilidade obtido, o robô é capaz de determinar em que estado este se encontra, estados como "via direta", "curva fechada à direita" ou "curva à esquerda". Após a detecção do estado, um comando de ação é enviado para o robô se mover no ambiente. Ao final dessa ação, uma nova imagem é capturada para que o todo o processo se repita. O diagrama da Figura 3.14(a) mostra

[2] `www.lrm.icmc.usp.br`

como funciona o sistema completo, note que a diferença entre esse diagrama e o diagrama da Figura 3.12 é que no trabalho de (Sales *et al.*, 2010) foi definido que a entidade "Navegação" é um autômato finito.



(a) Diagrama do Sistema de Navegação do Robô. Imagem adaptada de (Sales *et al.*, 2010).

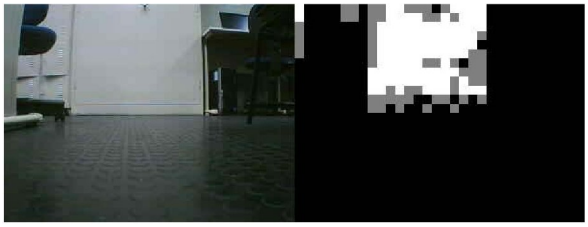
(b) Áreas de interesse no mapa de navegabilidade

Figura 3.14: Descrição do experimento utilizando o Sistema de Identificação de Superfícies Navegáveis proposto. Imagens retiradas de (Sales *et al.*, 2010).

(a) Via direta

(b) Curva 90° a esquerda

(c) Fim de Via se aproximando



(d) Fim de Via

Figura 3.15: Comparação de imagem original e do mapa de navegabilidade de um experimento utilizando o Sistema de Identificação de Superfícies Navegáveis proposto. Imagens retiradas de (Sales *et al.*, 2010).

O método de avaliação do mapa consiste em analisar áreas específicas como mostra a Figura 3.14(b). A região A0 é a região acima da linha do horizonte na imagem real e não afeta o resultado do autômato, por isso não é processada. As regiões A1, A2 e A3 são as áreas de inte-

resse e são analisadas da seguinte forma: se a maioria dos elementos dentro de A2 são iguais a "1" e os elementos dentro de A1 e A3 são iguais a "0" então tem-se uma via direta, ou seja, ação seguir em frente. Se A1 possuir mais elementos de valor "1" que A3 então o estado é "curva a esquerda", note que a região A2 ainda deve ter elementos com valor "1" caso contrário significa que a estrada acabou. Existem outros estados como curva para direita, e ajuste da rota para direita ou esquerda, mas o detalhamento desse sistema de controle está fora do escopo desse trabalho. A Figura 3.15 mostra a comparação da imagem original capturada pelo robô e o mapa de navegabilidade gerado de quatro estados diferentes.

De um modo geral, o sistema obteve bons resultados uma vez que o SRV conseguiu seguir trajetos complexos dentro de um ambiente adaptado para ele. Isto é, um ambiente real como uma sala onde o chão foi marcado com regiões para serem a "rua" por onde o robô pode andar.

3.3.2 Navegação Autônoma em Ambientes Urbanos Baseado em Casamento de Modelos

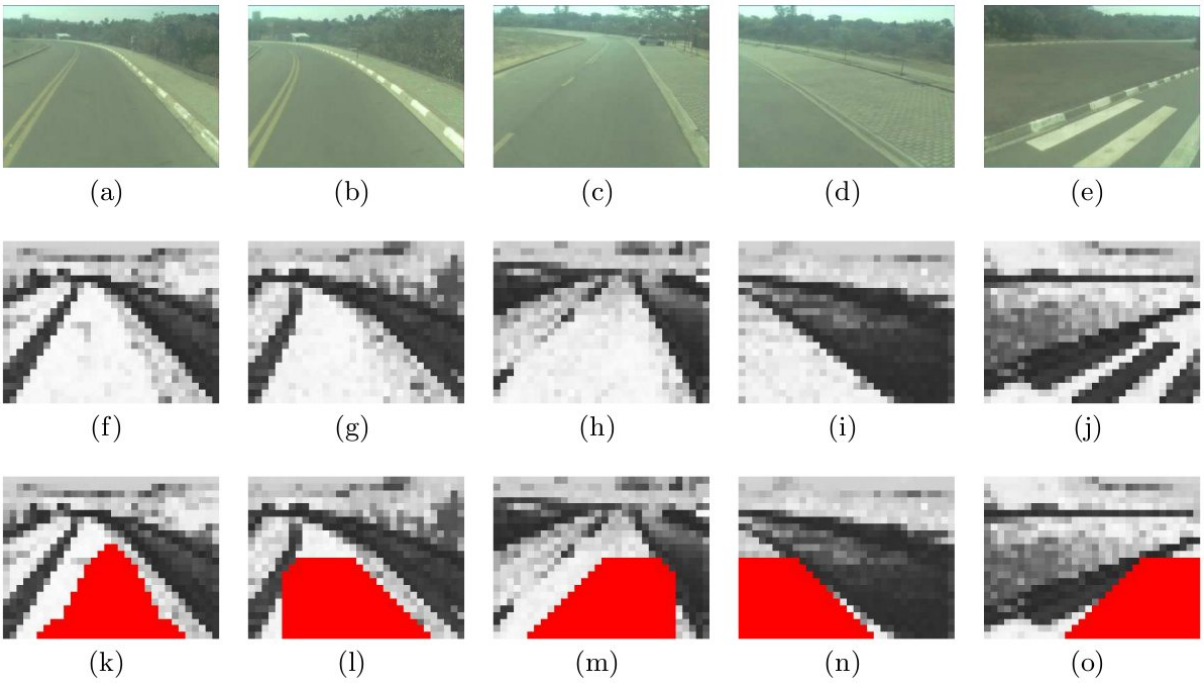


Figura 3.16: Experimento em ambiente externo utilizando o Sistema de Identificação de Superfícies Navegáveis. Imagens retiradas de (Souza *et al.*, 2011).

O método de (Souza *et al.*, 2011) assim como o de (Sales *et al.*, 2010) é baseado em um pós-processamento do mapa de navegabilidade gerado pelo nosso sistema de identificação para

determinar a ação a ser tomada pelo robô. Uma das principais diferenças entre os trabalhos é que (Souza *et al.*, 2011) utilizaram o veículo elétrico do ICMC para os testes em ambientes externos e urbanos.

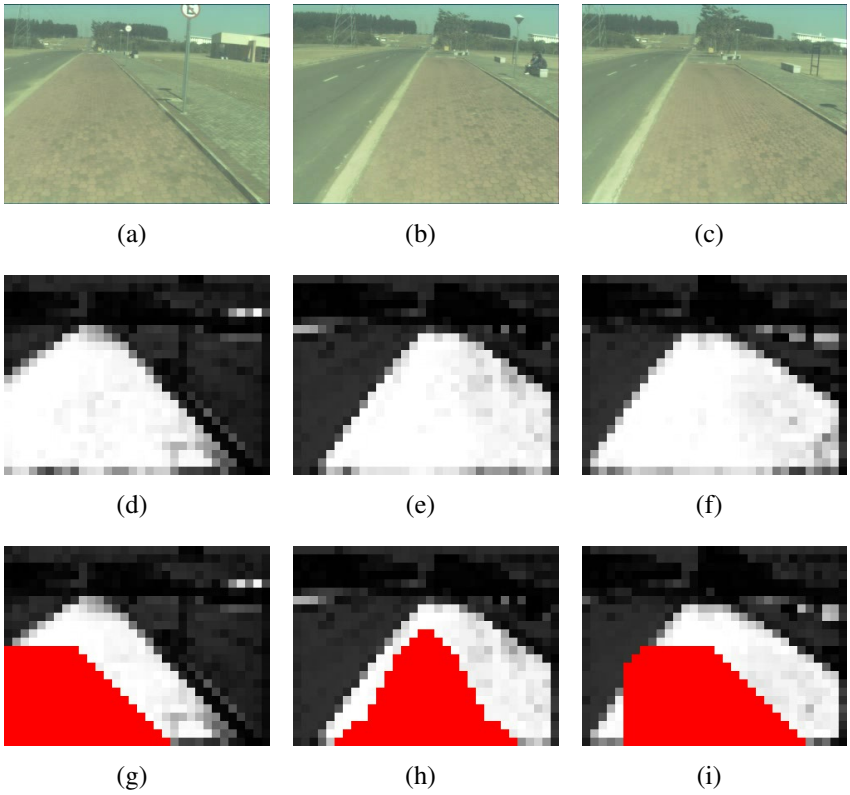


Figura 3.17: Experimento em ambiente externo em outro tipo de superfície. Mostra como o sistema é robusto, uma vez que ele consegue identificar a região navegável mesmo sendo diferente do asfalto.

Outra diferença do método de (Souza *et al.*, 2011) é que em seu pós-processamento, a classificação é um casamento de modelos (*"template-matching"*) de algumas máscaras sobre o mapa de navegabilidade visual. Essas máscaras possuem a forma geométrica trapezoidal que se aproxima da forma da rua em uma imagem. As Figuras 3.16(k) até 3.16(o) mostram as máscaras utilizadas em vermelho. Como o mapa possui vários valores entre "0" e "1", foi calculada a média dos graus de certeza dos blocos que fazem parte da máscara (região vermelha). A máscara que obtiver a maior média é a máscara que melhor corresponde a região navegável naquele momento. A partir desta máscara, o controlador decide qual o esterçamento que as rodas do carro devem manter para que ele continue dentro da pista. Sobre a Figura 3.16, as imagens (a) a (e) são as imagens reais obtidas pela câmera acoplada ao veículo elétrico. As imagens (f) até (j) são o resultado da identificação. As imagens (k) até (o) são as máscaras com

a melhor média para a imagem processada. Vale ressaltar que todas as máscaras são testadas em cada imagem, mas prevalece aquela com a maior média. Para mostrar a robustez do método, foi realizado o mesmo experimento em outra superfície como mostra a Figura 3.17. Note que desta vez, as RNA foram treinadas para permanecer dentro da região com ladrilhos vermelhos ao invés da região com asfalto cinza.

Além do *"template-matching"*, o método apresentado por (Souza *et al.*, 2011) usou um autômato finito determinístico (AFD) para reduzir a variação brusca entre os estados resultantes da máscara. Neste experimento a aceleração do veículo era controlada por um humano, mas o esterçamento das rodas foi totalmente autônomo. De um modo geral, este experimento obteve resultados aceitáveis e provou o bom funcionamento do sistema de identificação de superfícies navegáveis em ambientes externos.

3.4 Considerações

Neste capítulo, foi descrito o funcionamento do sistema de identificação de superfícies navegáveis baseado em redes neurais artificiais. Além do detalhamento de cada etapa, como a geração de atributos a partir de uma imagem e a classificação baseada em múltiplas RNA, foram comentados alguns trabalhos que utilizaram esses métodos a fim de exemplificar seu uso e comprovar seu funcionamento.

O Capítulo 4 traz uma sequência de experimentos isolados que visa mostrar a exatidão e o desempenho de detalhes descritos na metodologia proposta. Outros experimentos que deram suporte para a construção desta metodologia resultaram em publicações entre 2009-2010 (Apêndice A). Entretanto, estes foram apenas citados durante a descrição da metodologia a fim de não interferir no entendimento da proposta deste documento e complementar a base experimental deste trabalho.

CAPÍTULO

4

Experimentos e Resultados

A fim de validar a metodologia proposta, diversos experimentos foram realizados, os quais utilizaram os atributos listados no Apêndice B e cenários disponibilizados no Apêndice C. Mais especificamente, foram filmados vários trajetos percorridos por um veículo em diferentes condições de iluminação. Esses caminhos são compostos por ruas ladeadas de calçadas, estacionamentos, prédios e vegetação. Além disso, em alguns trechos, porções da rua apresentam condições adversas como sujeira e rastros de outros carros.

A coleta de dados foi realizada com um carro equipado com uma câmera digital Canon A610. No experimento com o veículo elétrico, foi utilizada a câmera VIDERE DCSG. A resolução das imagens dos vídeos utilizados é de (320×240) pixeis a 30 FPS. O carro e a câmera foram utilizados apenas para a coleta de dados. Todos os softwares foram desenvolvidos utilizando a linguagem C++ com a biblioteca FANN (FANN, 2010) para gerenciar as RNA e a biblioteca OpenCV (Bradski e Kaehler, 2008) para obter e exibir as imagens. O tamanho do bloco utilizado foi de $K = 10$, o que significa que cada imagem possui 768 blocos.

Este capítulo se divide em vários experimentos, onde na Seção 4.1 é comentado sobre um experimento que executa o método de seleção por saliências e mostra resultados gerados a partir dos atributos selecionados para mostrar sua coerência com o problema de identificação. Ainda nessa seção é descrito um estudo de desempenho de classificação de várias combinações de atributos de imagem para definir quais farão parte do sistema de identificação proposto. Na Seção 4.2 é discutido o método de avaliação baseado em FIT, comparando-o com o EQM e

HIT RATE. Na Seção 4.3 é apresentado o sistema de identificação de superfícies navegáveis em funcionamento, mostrando taxas de acerto e robustez obtida com o método proposto. Além disso, a partir dos resultados desse experimento foram definidos os parâmetros finais do sistema. Na Seção 4.4 é descrito sucintamente um experimento cujo objetivo é a detecção da linha do horizonte visando aumentar velocidade de processamento do sistema, esta seção foi criada apenas com o intuito de mostrar outras formas de aplicação desse sistema de identificação.

4.1 Seleção de Atributos

Os experimentos descritos nesta seção foram divididos em duas etapas. A primeira etapa utiliza o método de seleção de características baseado na medida de saliência, avaliado no trabalho de (dos Santos, 2007), para escolher os atributos que podem ter o melhor desempenho na classificação. Ainda nessa etapa é verificada a coerência dos resultados. A segunda etapa analisa combinações de atributos que são utilizadas como entrada de uma RNA para a classificação. A partir dessa análise, foram escolhidos os conjuntos de atributos das RNA utilizadas no sistema de identificação proposto nessa dissertação.

4.1.1 Seleção de Atributos Baseado na Saliência

O método de seleção de características baseado na medida de saliência foi utilizado para se estimar o grau de importância de cada atributo para cada cenário listado no Apêndice C. Para isso, foi gerada uma base de padrões de treinamento para cada cenário e uma outra base que contém todos os cenários que foi chamada de Cenário 9 ($C9$).

Esse método de seleção é dividido em duas fases, na primeira fase uma RNA é treinada utilizando o algoritmo de aprendizado *Resilient Propagation* até que sua taxa de erro chegue a um valor mínimo. Neste experimento o valor mínimo foi de 2% para todos os cenários exceto para os cenários 3, 4 e 9 que chegaram no máximo em 4% de erro. A topologia da RNA utilizada é 44 neurônios na camada de entrada, pois é o mesmo número de atributos avaliados, 5 neurônios na camada oculta e 2 neurônios na camada de saída, pois nosso problema de identificação possui apenas duas classes. De acordo com (dos Santos, 2007), esse número de neurônios na camada intermediária foi suficiente para suas RNA em seus experimentos alcançarem 100% de acerto e o número de neurônios na camada de saída deve ser igual ao número de classes. Desta maneira, apenas esse experimento utilizou uma RNA com a camada de saída com dois neurônios. Na segunda fase, são calculadas as saliências por meio dos pesos nas conexões entre os neurônios da RNA.

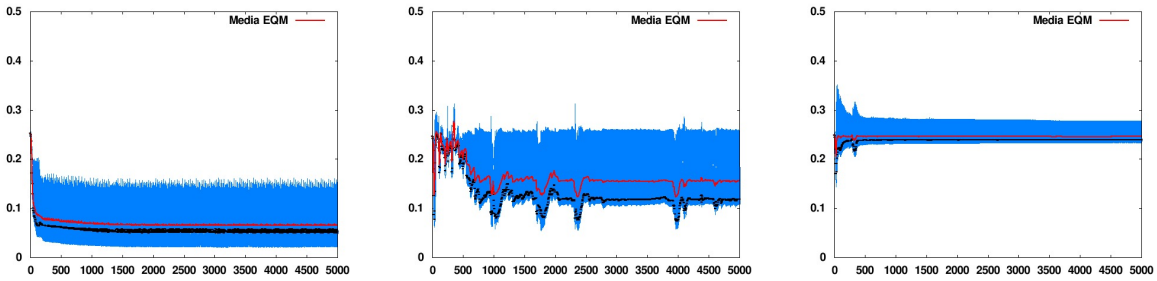
Tabela 4.1: Resultados do Método de Seleção por Saliências. Cada coluna representa um cenário e mostra uma nota entre 0 e 1 para cada atributo. Quanto maior essa nota maior a sua importância na classificação por RNA. Os resultados maiores que 0,5 estão destacados em cinza, onde quanto maior a nota mais escuro é a célula. A tabela está ordenada pela coluna "Média".

	Atributo	C1	C2	C3	C4	C5	C6	C7	C8	C9	Média	C{2,4,6,7,8}	C{1,3,5}
1	Média U	0,81	0,50	0,67	0,11	0,67	0,22	0,90	0,42	0,87	0,57	0,69	0,28
2	Média V	0,73	0,64	0,55	0,62	0,60	0,74	0,49	0,18	0,46	0,56	0,51	0,67
3	Média BNorm	0,82	0,15	0,70	0,12	0,61	0,35	0,58	0,56	0,47	0,48	0,65	0,21
4	Entropia H	0,33	0,77	0,39	0,69	0,28	0,81	0,14	0,21	0,37	0,44	0,27	0,75
5	Energia GNorm	0,30	0,76	0,44	0,82	0,24	0,52	0,26	0,18	0,44	0,44	0,29	0,70
6	Média H	0,42	0,16	0,76	0,18	0,41	0,57	0,45	0,42	0,53	0,43	0,49	0,30
7	Média GNorm	0,32	0,25	0,67	0,04	0,29	0,06	0,35	0,97	0,83	0,42	0,52	0,12
8	Média S	0,65	0,21	0,61	0,08	0,56	0,32	0,16	0,50	0,68	0,42	0,50	0,20
9	Média G	0,26	0,75	0,33	0,47	0,26	0,60	0,43	0,25	0,33	0,41	0,31	0,60
10	Média R	0,25	0,36	0,43	0,40	0,27	0,79	0,18	0,26	0,62	0,40	0,28	0,52
11	Entropia GNorm	0,50	0,31	0,40	0,21	0,37	0,16	0,27	0,30	0,57	0,34	0,37	0,23
12	Energia RNorm	0,26	0,37	0,37	0,28	0,22	0,34	0,25	0,22	0,30	0,29	0,26	0,33
13	Entropia R	0,40	0,33	0,52	0,20	0,26	0,25	0,24	0,12	0,18	0,28	0,31	0,26
14	Média B	0,47	0,49	0,34	0,26	0,18	0,26	0,08	0,06	0,31	0,27	0,22	0,34
15	Variância B	0,51	0,06	0,53	0,04	0,82	0,08	0,03	0,02	0,26	0,26	0,38	0,06
16	Entropia B	0,40	0,34	0,34	0,20	0,17	0,26	0,26	0,22	0,17	0,26	0,28	0,27
17	Energia BNorm	0,20	0,31	0,25	0,20	0,27	0,32	0,28	0,21	0,27	0,26	0,24	0,28
18	Entropia BNorm	0,21	0,36	0,28	0,26	0,28	0,19	0,33	0,23	0,15	0,25	0,27	0,27
19	Entropia Y	0,11	0,50	0,19	0,25	0,13	0,42	0,29	0,14	0,19	0,25	0,17	0,39
20	Entropia G	0,13	0,41	0,16	0,22	0,13	0,35	0,41	0,14	0,20	0,24	0,19	0,33
21	Média Y	0,16	0,47	0,13	0,43	0,11	0,44	0,16	0,04	0,18	0,24	0,12	0,44
22	Entropia S	0,23	0,32	0,15	0,35	0,20	0,35	0,17	0,13	0,19	0,23	0,17	0,34
23	Energia U	0,18	0,29	0,15	0,43	0,21	0,22	0,30	0,12	0,18	0,23	0,19	0,31
24	Entropia V	0,11	0,29	0,16	0,20	0,10	0,32	0,25	0,24	0,15	0,20	0,17	0,27
25	Energia H	0,16	0,28	0,13	0,17	0,17	0,27	0,18	0,22	0,16	0,19	0,17	0,24
26	Média RNorm	0,33	0,17	0,21	0,09	0,17	0,10	0,21	0,14	0,23	0,18	0,21	0,12
27	Energia S	0,13	0,30	0,13	0,34	0,10	0,29	0,11	0,11	0,13	0,18	0,11	0,31
28	Variância R	0,18	0,06	0,44	0,03	0,45	0,05	0,03	0,02	0,05	0,15	0,23	0,05
29	Energia Y	0,09	0,27	0,13	0,14	0,08	0,12	0,13	0,10	0,11	0,13	0,11	0,18
30	Entropia RNorm	0,12	0,15	0,20	0,14	0,11	0,13	0,07	0,09	0,08	0,12	0,12	0,14
31	Energia R	0,09	0,23	0,15	0,13	0,08	0,11	0,13	0,06	0,10	0,12	0,10	0,16
32	Energia G	0,09	0,24	0,09	0,13	0,09	0,12	0,13	0,07	0,09	0,12	0,09	0,16
33	Entropia U	0,13	0,11	0,17	0,11	0,13	0,13	0,08	0,05	0,11	0,11	0,11	0,11
34	Energia B	0,08	0,21	0,07	0,13	0,09	0,11	0,13	0,07	0,09	0,11	0,09	0,15
35	Energia V	0,11	0,18	0,08	0,11	0,08	0,10	0,12	0,06	0,09	0,10	0,09	0,13
36	Variância H	0,09	0,05	0,28	0,05	0,08	0,07	0,03	0,01	0,06	0,08	0,10	0,06
37	Variância U	0,04	0,07	0,11	0,05	0,17	0,08	0,03	0,01	0,04	0,07	0,07	0,07
38	Variância S	0,16	0,06	0,05	0,04	0,07	0,06	0,04	0,02	0,03	0,06	0,07	0,05
39	Variância V	0,05	0,06	0,11	0,03	0,03	0,05	0,03	0,02	0,04	0,05	0,05	0,05
40	Variância G	0,07	0,06	0,07	0,04	0,03	0,05	0,03	0,02	0,05	0,05	0,05	0,05
41	Variância Y	0,06	0,06	0,06	0,04	0,03	0,05	0,04	0,02	0,04	0,04	0,04	0,05
42	Variância RNorm	0,04	0,06	0,04	0,05	0,02	0,08	0,03	0,01	0,04	0,04	0,03	0,06
43	Variância GNorm	0,04	0,06	0,03	0,05	0,02	0,08	0,03	0,01	0,03	0,04	0,03	0,06
44	Variância BNorm	0,03	0,06	0,04	0,04	0,03	0,07	0,03	0,01	0,03	0,04	0,03	0,06

Para cada um dos nove cenários, o método baseado em saliência foi executado 30 vezes e a média dessas saliências resultantes para cada atributo são mostradas nas colunas $C1$ à $C9$ (Ci para o cenário i) da Tabela 4.1. Além disso, a tabela mostra a média das nove saliências de cada cenário na coluna "Média", a média dos cenários $\{2, 4, 6, 7, 8\}$ e por fim a média dos cenários $\{1, 3, 5\}$. A duas últimas colunas foram criadas no intuito de analisar a importância

dos atributos em diferentes condições de iluminação, pois os cenários {1, 3, 5} apresentam ocorrências de sombra durante o caminho percorrido enquanto os outros não. Os atributos da Tabela 4.1 estão ordenados em ordem decrescente pela coluna "Média".

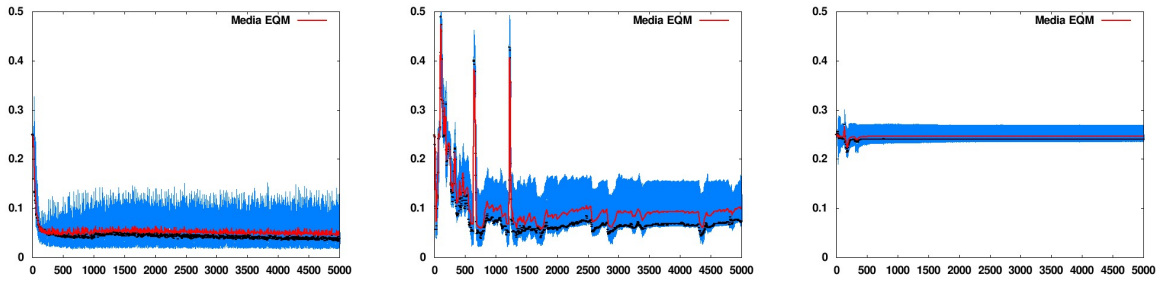
A fim de mostrar uma coerência entre os resultados do método de seleção baseado em saliências e o problema de identificação de superfícies navegáveis, foram selecionados arbitrariamente os atributos 1 a 5 da Tabela 4.1 para compor uma RNA chamada de RNA-1, os atributos 16 a 20 para compor a RNA-2 e os atributos 31 a 35 para compor a RNA-3. Como a seleção baseada em saliência identificou os atributos que tem mais importância na classificação pela RNA, os primeiros atributos listados na Tabela 4.1 são aqueles que melhor diferenciam os elementos navegáveis e não-navegáveis. Os cenários 1 e 2 foram escolhidos por se tratarem do mesmo percurso só que em condições de iluminação diferentes. Essas três RNA foram instanciadas 30 vezes, e para cada uma delas foi treinada até 5 mil ciclos para os dois cenários escolhidos. Os gráficos mostrados nas Figuras 4.1 e nas Figuras 4.2 mostram a relação do *EQM* $\times$ *CICLOS_DE_TREINAMENTO* das RNA com melhor desempenho dentre as 30 instâncias para os cenários 1 e 2 respectivamente. A linha vermelha é a média do EQM das imagens, o azul mostra o melhor e o pior EQM e o ponto preto é a mediana dentre os EQM obtidos naquele ciclo de treinamento.



(a) RNA-1 (b) RNA-2 (c) RNA-3

Figura 4.1: EQM das RNA-1, RNA-2 e RNA-3 para o Cenário 1, onde a linha vermelha é a média do EQM das imagens, o azul mostra o melhor e pior EQM e a linha preto é a mediana dentre os EQM obtidos.

Os resultados da RNA-1 convergiram rapidamente e mantiveram-se constantes muito próximos a um EQM de $0,05$. Os resultados da RNA-2, além de apresentarem um desempenho baixo, um EQM de quase $0,2$ para o Cenário 1 e um EQM próximo de $0,1$ para o Cenário 2, também apresentou um resultado oscilatório. Na topologia de RNA proposta, constatou-se que uma oscilação maior que $0,05$ é significativa, já que uma RNA começa com seu EQM em $0,25$ e termina, no melhor caso alcançado, com EQM próximo a $0,02$. A RNA-3 não obteve

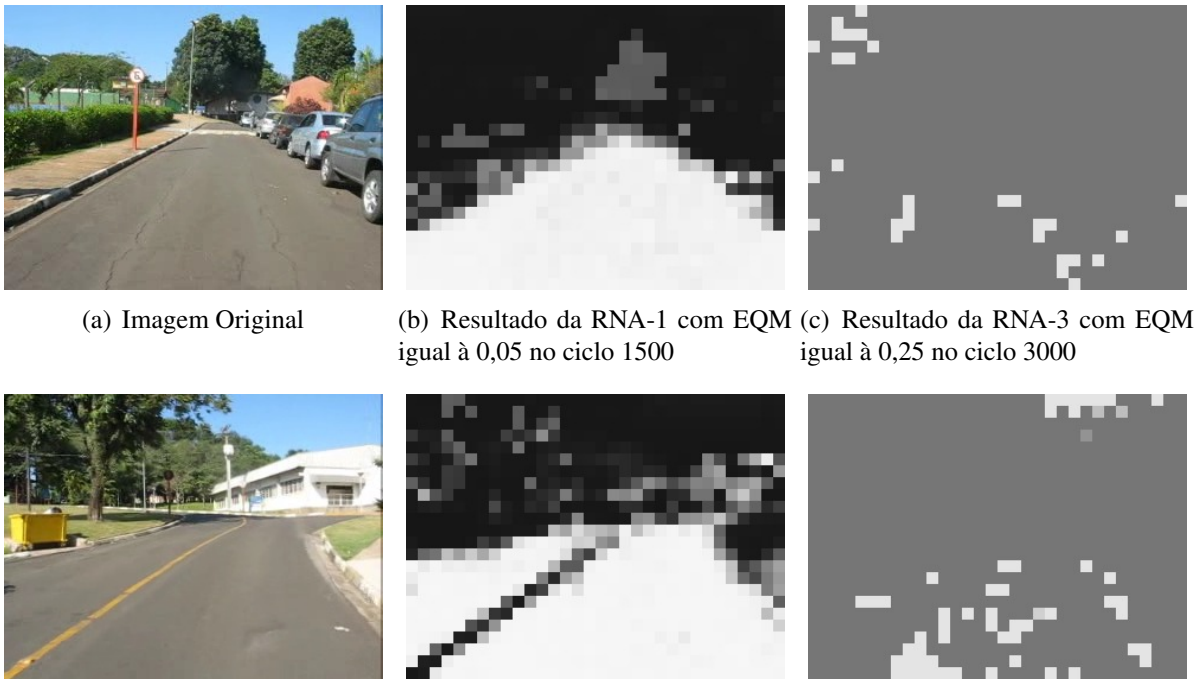


(a) RNA-1 (b) RNA-2 (c) RNA-3

Figura 4.2: EQM das RNA-1, RNA-2 e RNA-3 para o Cenário 2, onde a linha vermelha é a média do EQM das imagens, o azul mostra o melhor e pior EQM e a linha preto é a mediana dentre os EQM obtidos.

um desempenho de classificação aceitável, pois seu EQM se manteve na mesma taxa de $0,25$ desde o início do aprendizado. Outra constatação da topologia proposta é que um EQM igual à $0,25$ significa uma classificação muito ruim como mostram as Figuras 4.3(c) e 4.3(f), pois blocos navegáveis e blocos não navegáveis possuem a mesma classificação. Neste caso, quase todos os blocos receberam a nota $0,5$, o que significa que a RNA-3 possui total incerteza sobre a classificação dos blocos. Pode-se observar que os resultados são coerentes, pois a RNA-1 é a que obteve o melhor desempenho dentre as 3 RNA como mostra as Figuras 4.3(b) e 4.3(e), considerando a topologia proposta.

Concluindo, neste experimento pode-se observar a importância da escolha correta dos atributos de imagem, pois atributos ruins fazem com que a RNA obtenha resultados insatisfatórios ou possua um desempenho muito oscilatório. Apesar dos resultados apresentados, ainda não é possível afirmar que uma RNA com os primeiros atributos da Tabela 4.1 terá o melhor desempenho de classificação em todos os cenários. Como este trabalho se baseia na combinação dos resultados de várias RNA, é necessário definir vários conjuntos de atributos ao invés de apenas um. Além disso, de acordo com as medidas das saliências das colunas para cenários com sombra e cenários sem sombra pode-se notar que as condições influenciam diretamente o desempenho que os atributos obtêm. Devido a isso, uma outra análise foi realizada com o objetivo de selecionar os vários conjuntos de atributos e saber quando estes possuem um bom desempenho. Essa análise é descrita na próxima seção.



(a) Imagem Original (b) Resultado da RNA-1 com EQM igual à 0,05 no ciclo 1500 (c) Resultado da RNA-3 com EQM igual à 0,25 no ciclo 3000

(d) Imagem Original (e) Resultado da RNA-1 com EQM igual à 0,05 no ciclo 2000 (f) Resultado da RNA-3 com EQM igual à 0,25 no ciclo 2000

Figura 4.3: Resultados das RNA que obtiveram o melhor e pior desempenho. As imagens (c) e (f) mostram que a RNA-3 não diferencia os blocos navegáveis dos não-navegáveis, pois quase todos os blocos receberam a nota $0,5$, o que significa que a RNA possui total incerteza sobre a classe a qual o bloco pertence. As imagens (b) e (e) mostram os bons resultados da RNA-1, pois grande parte da região navegável foi classificada com um grau de certeza muito elevado por isso os blocos pintados de branco.

4.1.2 Análise de Desempenho das RNA com Diferentes Atributos de Entrada

A partir dos resultados mostrados na Tabela 4.1 foram geradas três listas de prioridades de atributos. A lista 1 possui os atributos ordenados em ordem decrescente pela coluna "Média", a lista 2 possui os atributos ordenados pela coluna "C{2,4,6,7,8}" e a lista 3 pela coluna "C{1,3,5}". Baseado nessas listas foram criados três conjuntos de atributos que são os 10 melhores candidatos de cada lista. Esses atributos podem ser vistos na Tabela 4.2 e de acordo com experimento realizado com o seletor por saliências são os que melhor diferenciam regiões navegáveis de não-navegáveis para diferentes situações. O experimento desta seção analisou o desempenho de várias RNA baseadas nesses atributos escolhidos.

Tabela 4.2: Atributos candidatos aos melhores desempenhos. Três listas de atributos onde cada uma foi gerada a partir dos resultados sobre cenários em diferentes condições de iluminação.

(T1) Média Geral	
1	Média U
2	Média V
3	Média BNorm
4	Entropia H
5	Energia GNorm
6	Média H
7	Média GNorm
8	Média S
9	Média G
10	Média R

(T2) C{2,4,6,7,8}	
1	Média V
2	Entropia H
3	Energia GNorm
4	Média G
5	Média U
6	Média R
7	Média H
8	Média BNorm
9	Média GNorm
10	Entropia Y

(T3) C{1,3,5}	
1	Média U
2	Média BNorm
3	Média V
4	Variância B
5	Média S
6	Média H
7	Média GNorm
8	Entropia GNorm
9	Entropia R
10	Variância R

Baseado no resultados do experimento desta seção foi possível determinar relativamente quantos atributos são necessários para que uma RNA obtenha um bom desempenho de classificação. Entende-se em obter uma boa classificação, uma RNA que converge rapidamente e alcança um EQM baixo (aproximadamente $0,05$ para o conjunto de dados em questão). Outra observação foi em quais situações (iluminação constante ou ocorrências de sombra) um conjunto de atributos possui bom desempenho de classificação.

As topologias testadas foram criadas de acordo com os candidatos da Tabela 4.2. Para a Tabela "Média Geral" foram criadas 10 topologias de RNA que diferem apenas na camada de entrada, onde a primeira contem apenas o primeiro atributo, a segunda possui os 2 primeiros atributos, a terceira possui os 3 primeiros atributos, e assim por diante até a décima topologia de RNA que possui os 10 atributos da tabela. O mesmo aconteceu para as demais tabelas, totalizando as 30 topologias de RNA do experimento. A camada intermediária de todas as RNA possui 5 neurônios, e a de saída possui 1 neurônio. Cada uma dessas topologias foram instanciadas 30 vezes, onde cada uma delas foi treinada até 5 mil ciclos para cada um dos 8 cenários do Apêndice C.

Os resultados desse experimento são mostrados na Tabela 4.3, onde as linhas estão divididas em 3 grandes grupos (T1, T2 e T3) que representam os atributos contidos dentro de cada uma das Tabelas 4.2. Os números das linhas dentro de cada grupo representam o número de atributos utilizados. Para cada topologia de RNA, é mostrado o melhor EQM alcançado dentre as 30 instâncias para os cenários apresentados nas colunas. Nota-se também que cada cenário possui duas colunas "M" e "F", onde a coluna "M" é o menor EQM alcançado durante todo o treinamento e a coluna "F" é o menor EQM no ciclo de treinamento 5000.

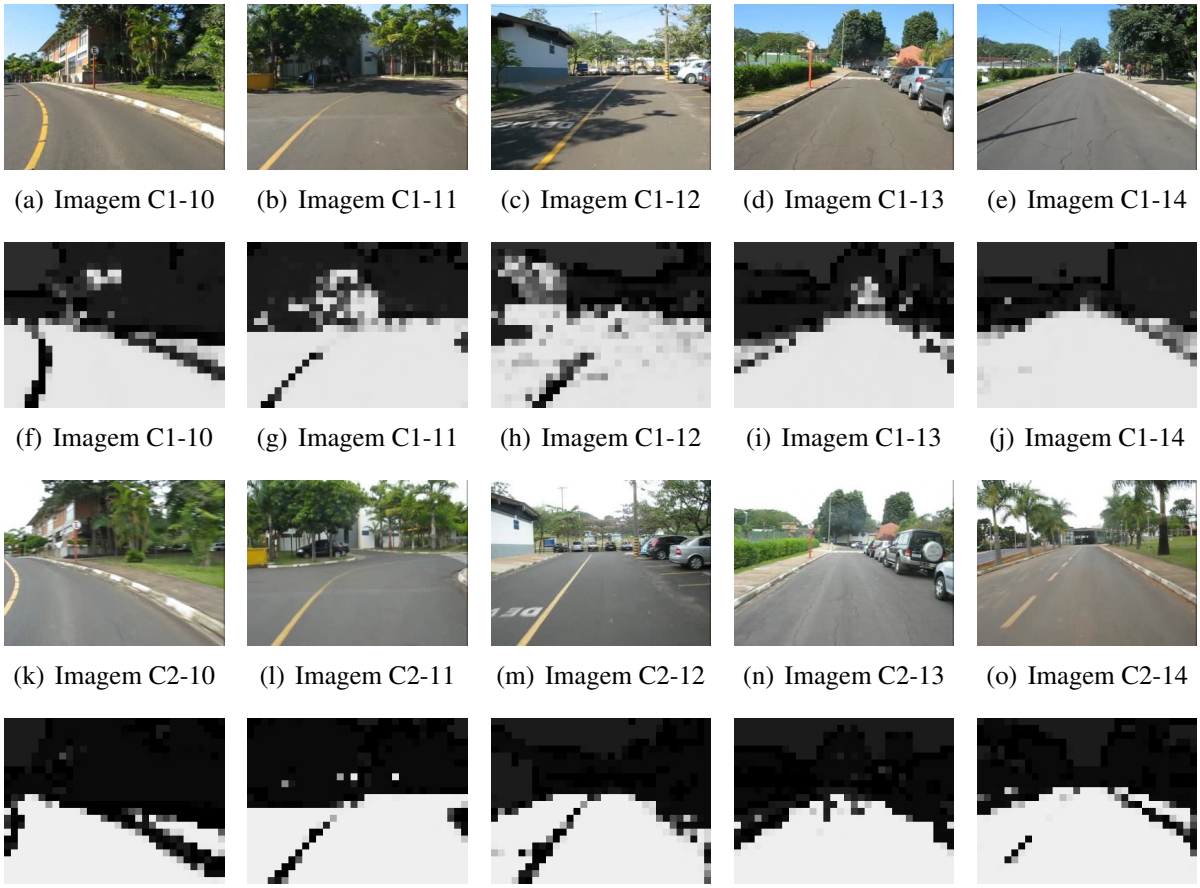
Como visto anteriormente, um EQM menor ou igual à $0,05$ pode ter um bom desempenho na classificação. Devido a isso, foram discutidas apenas as RNA com EQM abaixo desse valor. Na Tabela 4.3, os EQM abaixo de $0,03$ foram coloridos de cinza escuro e as demais menores que

	Número de Atributos	Cenários															
		C1		C2		C3		C4		C5		C6		C7		C8	
		M	F	M	F	M	F	M	F	M	F	M	F	M	F	M	F
T1	1	0,1400	0,1433	0,1873	0,1920	0,2145	0,2180	0,2356	0,2458	0,1234	0,1764	0,1852	0,1947	0,0987	0,0987	0,1638	0,1978
	2	0,1126	0,1154	0,1373	0,1457	0,2012	0,2092	0,1488	0,1515	0,1113	0,1190	0,1166	0,1474	0,0813	0,1006	0,1465	0,1596
	3	0,0764	0,0868	0,0815	0,0815	0,1183	0,1332	0,0806	0,1038	0,0893	0,1063	0,0803	0,1072	0,0732	0,0893	0,0724	0,1210
	4	0,0370	0,0479	0,0394	0,0457	0,0777	0,0822	0,0397	0,0429	0,0453	0,0523	0,0314	0,0332	0,0497	0,0564	0,0461	0,0546
	5	0,0293	0,0299	0,0281	0,0289	0,0600	0,0638	0,0235	0,0260	0,0325	0,0377	0,0232	0,0247	0,0462	0,0492	0,0360	0,0439
	6	0,0257	0,0277	0,0273	0,0285	0,0564	0,0603	0,0244	0,0258	0,0261	0,0334	0,0195	0,0210	0,0455	0,0474	0,0232	0,0340
	7	0,0233	0,0267	0,0219	0,0263	0,0564	0,0569	0,0234	0,0263	0,0244	0,0264	0,0146	0,0194	0,0452	0,0472	0,0252	0,0301
	8	0,0198	0,0221	0,0170	0,0184	0,0553	0,0631	0,0187	0,0217	0,0237	0,0252	0,0122	0,0150	0,0440	0,0459	0,0320	0,0320
	9	0,0208	0,0233	0,0171	0,0215	0,0569	0,0652	0,0201	0,0225	0,0235	0,0251	0,0114	0,0154	0,0440	0,0471	0,0262	0,0402
	10	0,0205	0,0238	0,0158	0,0179	0,0559	0,0568	0,0194	0,0219	0,0226	0,0259	0,0133	0,0157	0,0445	0,0492	0,0273	0,0386
T2	1	0,1564	0,1582	0,1500	0,1627	0,2071	0,2080	0,1461	0,1514	0,1930	0,1969	0,1633	0,1677	0,1069	0,1151	0,1684	0,1747
	2	0,0968	0,0980	0,0759	0,0774	0,1476	0,1491	0,0696	0,0697	0,1075	0,1097	0,0722	0,0740	0,0565	0,0688	0,0922	0,0979
	3	0,0394	0,0405	0,0276	0,0288	0,0731	0,0746	0,0270	0,0270	0,0378	0,0392	0,0240	0,0249	0,0496	0,0514	0,0439	0,0482
	4	0,0428	0,0435	0,0318	0,0346	0,0762	0,0773	0,0290	0,0290	0,0350	0,0362	0,0252	0,0258	0,0495	0,0521	0,0485	0,0558
	5	0,0336	0,0365	0,0305	0,0308	0,0706	0,0706	0,0259	0,0266	0,0274	0,0307	0,0234	0,0239	0,0463	0,0533	0,0435	0,0468
	6	0,0264	0,0272	0,0282	0,0289	0,0674	0,0718	0,0218	0,0246	0,0290	0,0326	0,0153	0,0170	0,0457	0,0467	0,0414	0,0419
	7	0,0233	0,0261	0,0274	0,0283	0,0604	0,0618	0,0246	0,0264	0,0253	0,0275	0,0208	0,0233	0,0433	0,0465	0,0337	0,0375
	8	0,0241	0,0249	0,0257	0,0277	0,0546	0,0619	0,0211	0,0267	0,0236	0,0261	0,0187	0,0219	0,0437	0,0484	0,0196	0,0315
	9	0,0239	0,0255	0,0240	0,0245	0,0599	0,0620	0,0227	0,0251	0,0247	0,0259	0,0138	0,0140	0,0439	0,0460	0,0280	0,0359
	10	0,0197	0,0226	0,0127	0,0135	0,0534	0,0574	0,0149	0,0166	0,0188	0,0226	0,0053	0,0069	0,0167	0,0209	0,0161	0,0187
T3	1	0,1401	0,1429	0,1873	0,1928	0,2147	0,2156	0,2490	0,2497	0,2127	0,2142	0,1828	0,1828	0,0956	0,0962	0,1598	0,1598
	2	0,0916	0,0941	0,1381	0,1790	0,1355	0,1355	0,1393	0,1497	0,0918	0,1006	0,1467	0,1672	0,0855	0,0992	0,1006	0,1517
	3	0,0607	0,0639	0,0812	0,1275	0,1181	0,1240	0,0814	0,0826	0,0947	0,1064	0,0926	0,1151	0,0657	0,0726	0,0724	0,0966
	4	0,0945	0,1009	0,0291	0,0316	0,1189	0,1350	0,0368	0,0396	0,0916	0,1015	0,0225	0,0263	0,0275	0,0313	0,0273	0,0385
	5	0,0706	0,0972	0,0246	0,0246	0,1013	0,1169	0,0296	0,0570	0,0374	0,0523	0,0088	0,0095	0,0279	0,0292	0,0350	0,0378
	6	0,0409	0,0433	0,0171	0,0198	0,0726	0,0800	0,0161	0,0170	0,0533	0,0774	0,0301	0,0326	0,0226	0,0323	0,0290	0,0310
	7	0,0605	0,0883	0,0140	0,0165	0,0950	0,1177	0,0212	0,0231	0,0312	0,0346	0,0091	0,0106	0,0220	0,0315	0,0247	0,0278
	8	0,0200	0,0217	0,0135	0,0148	0,0539	0,0571	0,0156	0,0164	0,0279	0,0312	0,0093	0,0116	0,0202	0,0249	0,0252	0,0295
	9	0,0565	0,0739	0,0133	0,0140	0,0563	0,0574	0,0185	0,0248	0,0269	0,0269	0,0123	0,0504	0,0207	0,0249	0,0311	0,0339
	10	0,0546	0,0745	0,0184	0,0188	0,0516	0,0527	0,0148	0,0152	0,0261	0,0359	0,0082	0,0108	0,0210	0,0320	0,0299	0,0324

Tabela 4.3: Resultados dos EQM para seleção de atributos que serão utilizados no sistema. Para cada cenário a coluna "M" mostra o melhor desempenho alcançado e a coluna "F" o desempenho da RNA no último ciclo de treinamento. As células com erros menores que $0,03$ foram pintados de cinza escuro, as células com EQM menor que $0,05$ foram pintadas de cinza claro.

$0,05$ com tons de cinza mais claros. Essa apresentação da tabela deixou evidente o desempenho muito parecido entre os grupos T1 e T2. Tal semelhança era esperada uma vez que muitos dos seus atributos são iguais. Pode-se notar também que dentre todos os cenários, o C3 é o mais complexo, pois nenhuma das RNA avaliadas conseguiu EQM menor que $0,05$, tal complexidade pode ser atribuída as muitas ocorrências de sombras no percurso.

Observando o grupo T1 isoladamente, pode-se verificar que uma RNA com os seis primeiros atributos consegue um EQM menor que $0,03$ em seis dos oito cenários testados. Apenas os cenários 3 e 7 obtiveram EQM maior, e mesmo assim obtiveram desempenhos muito próximos a $0,05$. Pode-se concluir que o uma RNA com estes seis atributos possui um bom desempenho em sua classificação na maioria dos cenários. Pode-se ver também que esse conjunto obtêm bons desempenhos a partir de RNA com quatro atributos, mas é com seis atributos que passa a obter resultados bem mais significativos.



(a) Imagem C1-10 (b) Imagem C1-11 (c) Imagem C1-12 (d) Imagem C1-13 (e) Imagem C1-14

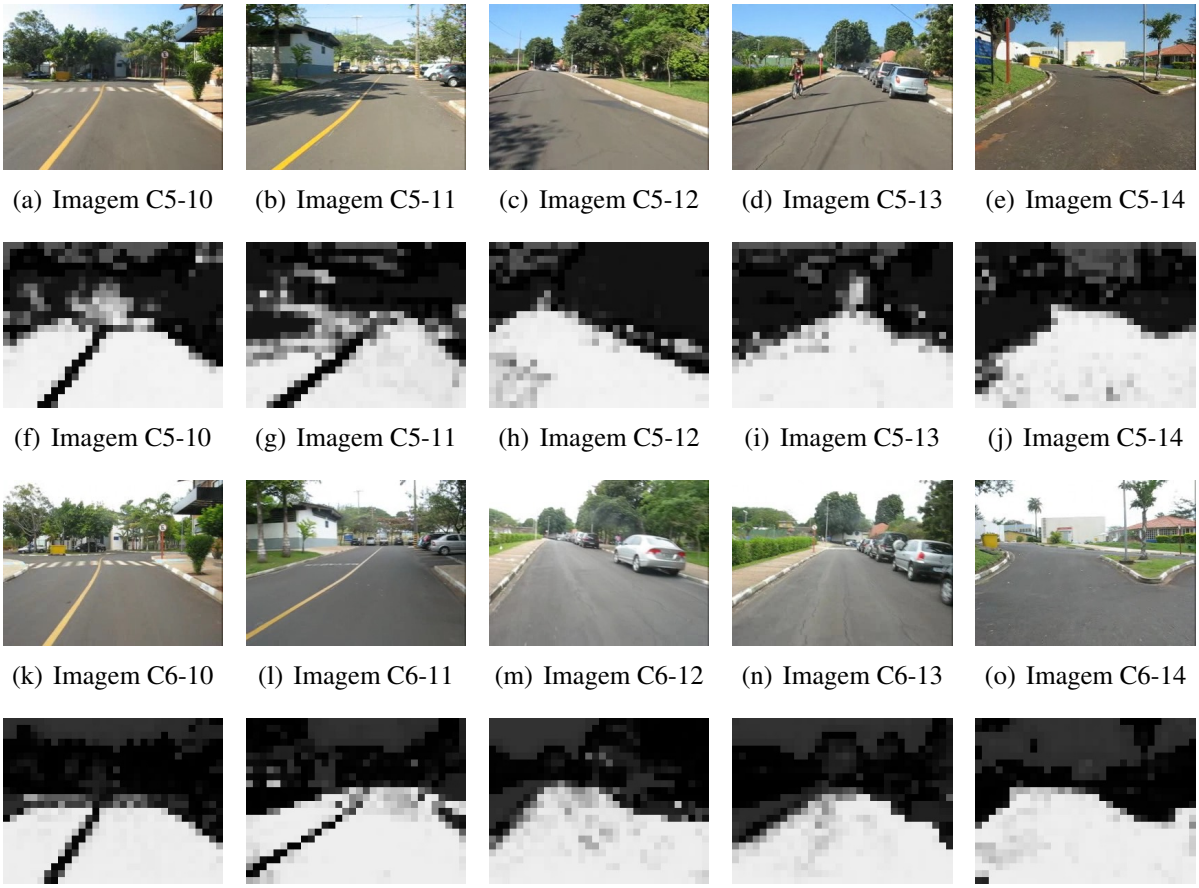
(f) Imagem C1-10 (g) Imagem C1-11 (h) Imagem C1-12 (i) Imagem C1-13 (j) Imagem C1-14

(k) Imagem C2-10 (l) Imagem C2-11 (m) Imagem C2-12 (n) Imagem C2-13 (o) Imagem C2-14

(p) Imagem C2-10 (q) Imagem C2-11 (r) Imagem C2-12 (s) Imagem C2-13 (t) Imagem C2-14

Figura 4.4: Resultados nos cenários 1 e 2 obtidos da RNA do grupo T2 com 10 atributos de imagem.

No grupo T2, seis atributos garantem bom desempenho em cinco cenários, apenas para os cenários 3, 7 e 8 obtiveram EQM maior que $0,03$. E assim como o grupo T1, nesses outros três cenários, seu desempenho ficou próximo a $0,05$. Um detalhe importante sobre este grupo, é que quando foram utilizados seus dez atributos, sete dos oito cenários passaram a ter um EQM abaixo de $0,023$, o que representa um ótimo desempenho. A Figura 4.4, Figura 4.5 e Figura 4.6 mostram vários resultados de RNA que utilizaram esse grupo de atributos.



(a) Imagem C5-10 (b) Imagem C5-11 (c) Imagem C5-12 (d) Imagem C5-13 (e) Imagem C5-14

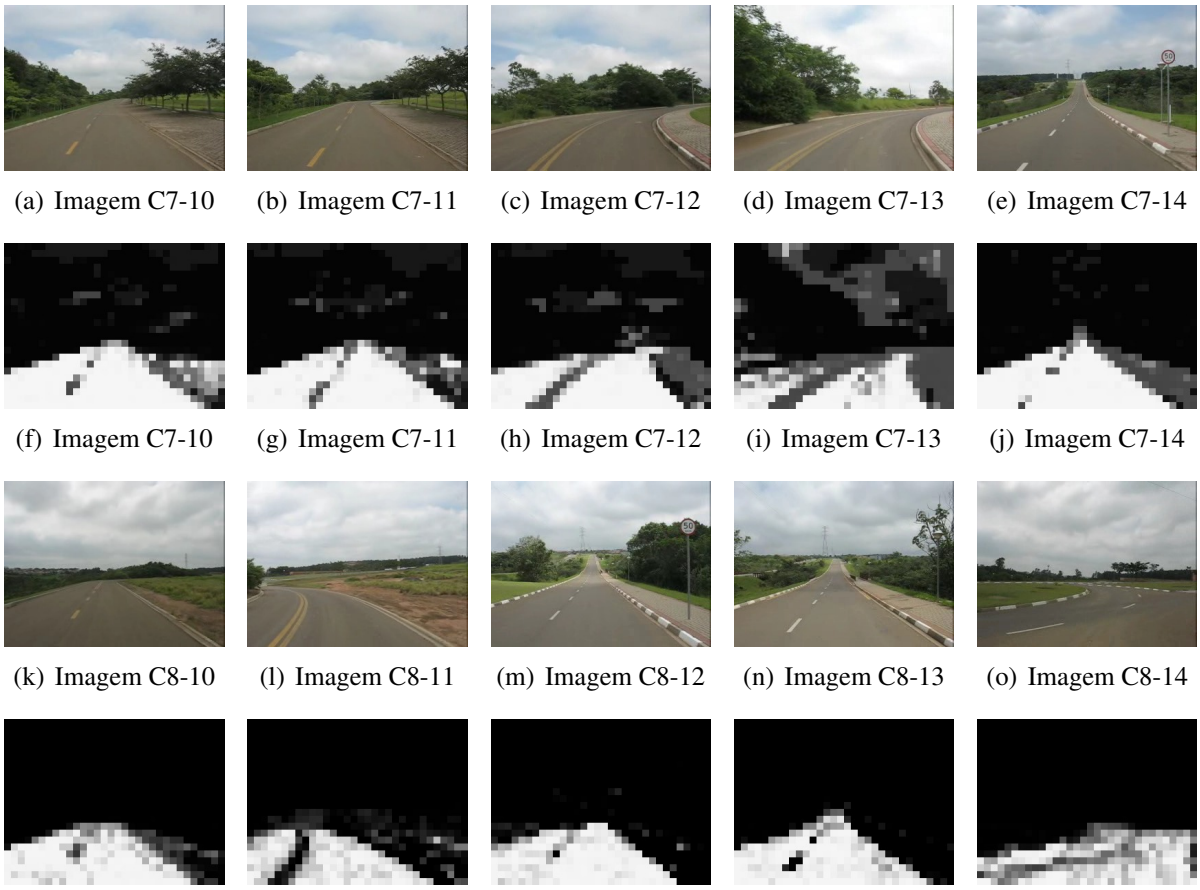
(f) Imagem C5-10 (g) Imagem C5-11 (h) Imagem C5-12 (i) Imagem C5-13 (j) Imagem C5-14

(k) Imagem C6-10 (l) Imagem C6-11 (m) Imagem C6-12 (n) Imagem C6-13 (o) Imagem C6-14

(p) Imagem C6-10 (q) Imagem C6-11 (r) Imagem C6-12 (s) Imagem C6-13 (t) Imagem C6-14

Figura 4.5: Resultados nos cenários 5 e 6 obtidos da RNA do grupo T2 com 10 atributos de imagem.

A partir dos resultados do grupo T3, pode-se notar que uma RNA com os oito primeiros atributos consegue um bom desempenho em seis cenários. É importante ressaltar que diferentemente dos dois outros grupos, o grupo T3 obteve bons resultados com apenas 4 atributos nos cenários 7 e 8. Além disso, esse grupo de atributos obteve o melhor desempenho alcançado no Cenário 3 (cenário considerado como o mais complexo dentre os cenários avaliados).



(a) Imagem C7-10 (b) Imagem C7-11 (c) Imagem C7-12 (d) Imagem C7-13 (e) Imagem C7-14

(f) Imagem C7-10 (g) Imagem C7-11 (h) Imagem C7-12 (i) Imagem C7-13 (j) Imagem C7-14

(k) Imagem C8-10 (l) Imagem C8-11 (m) Imagem C8-12 (n) Imagem C8-13 (o) Imagem C8-14

(p) Imagem C8-10 (q) Imagem C8-11 (r) Imagem C8-12 (s) Imagem C8-13 (t) Imagem C8-14

Figura 4.6: Resultados nos cenários 7 e 8 obtidos da RNA do grupo T2 com 10 atributos de imagem.

É importante ressaltar que esses desempenhos foram calculados a partir de uma RNA treinada com padrões de um determinado cenário e avaliada com outros padrões do mesmo cenário não apresentados durante o treinamento. Baseado nas observações deste experimento, foram definidos três grupos de atributos de imagem para compor o nosso sistema de identificação. Estes atributos estão listados na seguir:

- Seis atributos de T1: Média U, Média V, Média B Normalizado, Entropia H, Energia G Normalizado e Média H
- Dez atributos de T2: Média V, Entropia H, Energia G Normalizado, Média G, Média U, Média R, Média H, Média B Normalizado, Média G Normalizado e Entropia Y.
- Oito atributos de T3: Média U, Média B Normalizado, Média V, Variância B, Média S, Média H, Média G Normalizado, Entropia G Normalizado.

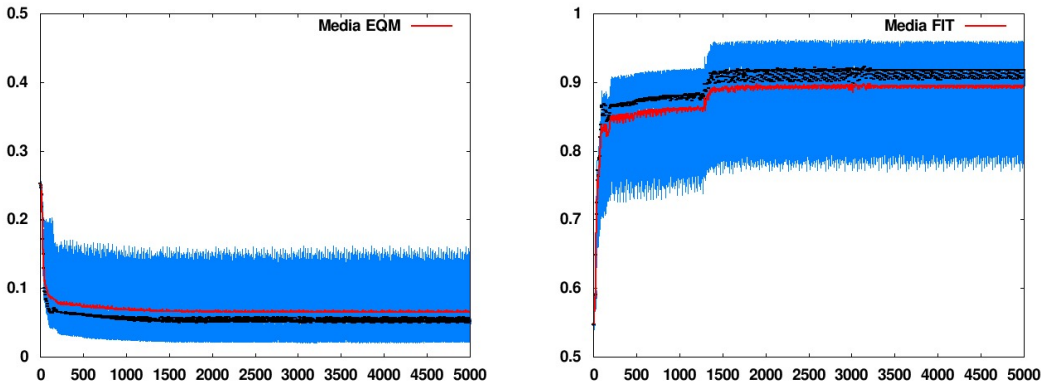
Em relação ao grupo T2 poderiam ser selecionados apenas os 6 primeiros atributos, entretanto utilizando os 10 tem-se um bom desempenho em quase todos os cenários. Desta forma, esse grupo é um forte candidato a ter a melhor generalização nos cenários apresentados. A classificação de uma RNA que utiliza o grupo T2 associada as demais classificações de RNA que utilizam os outros grupos significa em um aumento no desempenho geral (em todos os cenários) da classificação do que se fosse utilizada apenas uma única RNA.

4.2 Método de Avaliação

O método de avaliação por FIT foi um método que utiliza uma métrica definida com o objetivo de relacionar as informações de erro médio e precisão com o grau de certeza que uma RNA possui sobre um determinado padrão. Ele foi desenvolvido para dar maior importância aos padrões classificados erroneamente onde a diferença entre o resultado esperado e o obtido é muito pequena. Pois desta forma, diferencia-se uma RNA que errou poucos padrões com essa diferença bem alta da RNA que errou muitos padrões com uma diferença muito baixa. Em outras palavras, é possível distinguir a RNA que possui muitos padrões quase classificados corretamente da RNA que errou drasticamente poucos padrões apesar de terem a mesma taxa de erro médio.

Seguindo a metodologia descrita na Seção 3.2.2 foi realizado um experimento com o objetivo de validar essa metodologia de avaliação. Para isto, 30 RNAs foram instanciadas de 30 diferentes conjuntos de atributos de entrada (30 topologias de RNA diferentes) para todos os cenários do Apêndice C. A diferença entre os métodos convencionais (EQM, HIT RATE) e esse método (FIT) pode ser vista em vários resultados desse experimento. É importante ressaltar que o FIT é uma nota que representa o acerto, ou seja, a nota $1,0$ representa o acerto máximo e a nota $0,0$ representa a pior classificação possível que é quando o classificador inverte as classificações. A principal contribuição deste método é mostrada na Figura 4.7, onde nesse caso, os padrões foram gerados do Cenário 1 e a RNA possui os 5 primeiros neurônios da Tabela 4.1.

Analisando os gráficos da Figura 4.7 pode-se notar que o EQM indica uma convergência desde 500 ciclos de treinamento enquanto o FIT ainda mostra uma melhoria no aprendizado entre os ciclos 1000 e 1500. A Figura 4.8 mostra 3 exemplos dessa pequena melhoria entre os resultados do ciclo 1000 e do ciclo 1500. Para cada figura, a imagem a esquerda é a original captura pela câmera, a imagem central é o resultado da classificação da RNA no ciclo de treinamento 1000 enquanto a imagem da direita é o resultado da classificação da RNA no ciclo de treinamento 1500. Apesar das imagens serem visualmente muito similares, o resultado da RNA no ciclo 1500 retorna cores mais escuras para os blocos do céu.



(a) EQM - Taxa de erro (b) FIT - Taxa de acerto

Figura 4.7: Experimento no Cenário 1. Diferença entre as avaliações EQM e FIT. Pode-se ver que o EQM indica uma convergência enquanto o FIT ainda mostra uma melhoria no aprendizado entre os ciclos 1000 e 1500.

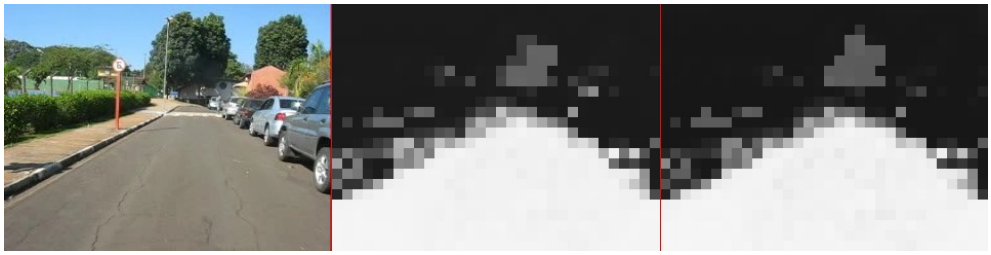
A partir dessa observação, pode-se concluir que a classificação desses blocos ficou mais próxima de "0". O que significa um aumento da certeza sobre a não-navegabilidade desses blocos durante o treinamento da RNA. Para fundamentar essa conclusão foram calculados os *HIT RATE* nos ciclos 1000 e 1500. A Tabela 4.4 mostra esses valores, onde na primeira coluna temos a taxa de acerto se a precisão for igual a $0,1$ e a segunda coluna mostra a taxa de acerto para a precisão for $0,2$. Visualizando as diferenças entre as linhas do ciclo 1000 e 1500 na coluna de precisão $0,1$, pode-se perceber um significativo aumento na taxa de acerto. Como dito anteriormente, esse aumento é resultante da classificação mais próxima de "0" dos blocos. Ou seja, 30% das classificações no ciclo 1000 estavam próximas de $0,1$ mas ainda eram maiores, não sendo contabilizadas na taxa de acerto. Após mais 500 ciclos de treinamento, o resultados dessas classificações passaram a ser menores ou iguais a $0,1$, sendo contabilizado na taxa de acerto. Apesar dessa melhoria na taxa de acerto, o EQM não se alterou. Isso se deve ao fato que outros blocos passaram a ter uma classificação pior, mas que, de acordo com os resultados mostrados na Tabela 4.4, não foram suficientes para alterar o resultado final.

	Cenário 1	
Ciclos de treinamento	precisão 0,1	precisão 0,2
1000 ciclos	37%	77,23%
1500 ciclos	66,38%	77,35%

Tabela 4.4: Taxa de acerto da RNA para os ciclos 1000 e 1500 no Cenário 1.

Esse comportamento acontece em vários outros casos, como em uma instância dessa mesma RNA treinada para o Cenário 2 (Figura 4.9 e Figura 4.10), onde as diferenças são muito mais

(a) Imagem 10 do Cenário 1



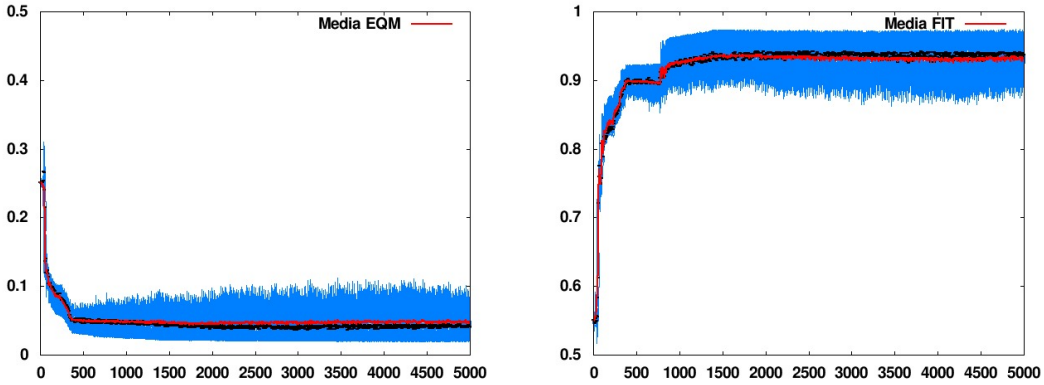
(b) Imagem 13 do Cenário 1

(c) Imagem 14 do Cenário 1

Figura 4.8: Experimento no Cenário 1. Comparação dos resultados entre os ciclos 1000 e 1500. Pode-se ver que a certeza sobre a não navegabilidade dos blocos superiores aumenta conforme a RNA é treinada.

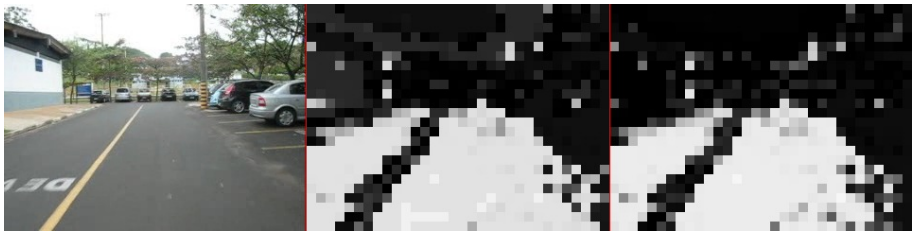
aparentes. Assim como na Figura 4.8, a imagem a esquerda é a original obtida pela câmera, a do meio é um resultado da RNA no ciclo de treinamento 500 e a imagem mais a direita é o resultado da classificação da RNA no ciclo de treinamento 1000. Pode-se observar facilmente que o grau de certeza sobre a não-navegabilidade do céu aumentou substancialmente. Quando não acontece a situação onde o FIT melhora e o EQM se mantém constante, a função FIT se comporta igualmente ao EQM, a Figura 4.11 mostra o caso em que ambos convergem.

Com base nos experimentos realizados foi constatado que o método de avaliação por FIT é mais adequado que o EQM em algumas situações e possui o mesmo desempenho que o EQM nos demais casos. Devido a essas conclusões, o sistema de identificação baseia-se nessa avaliação para determinar quando encerrar o treinamento de cada RNA antes de iniciar o uso do sistema na identificação da região navegável no ambiente.

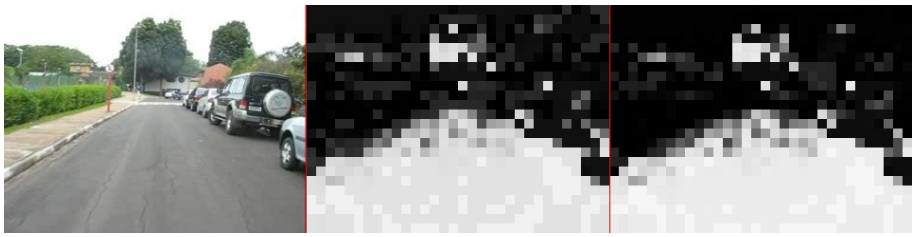


(a) EQM - Taxa de erro (b) FIT - Taxa de acerto

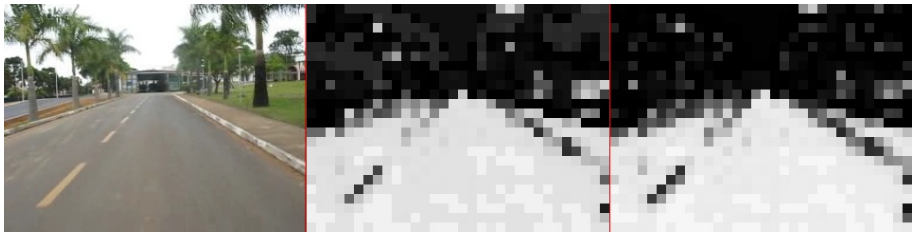
Figura 4.9: Experimento no Cenário 2. Diferença entre as avaliações EQM e FIT. Pode-se ver que o EQM indica uma convergência enquanto o FIT ainda mostra uma melhoria no aprendizado.



(a) Imagem 12 do Cenário 2

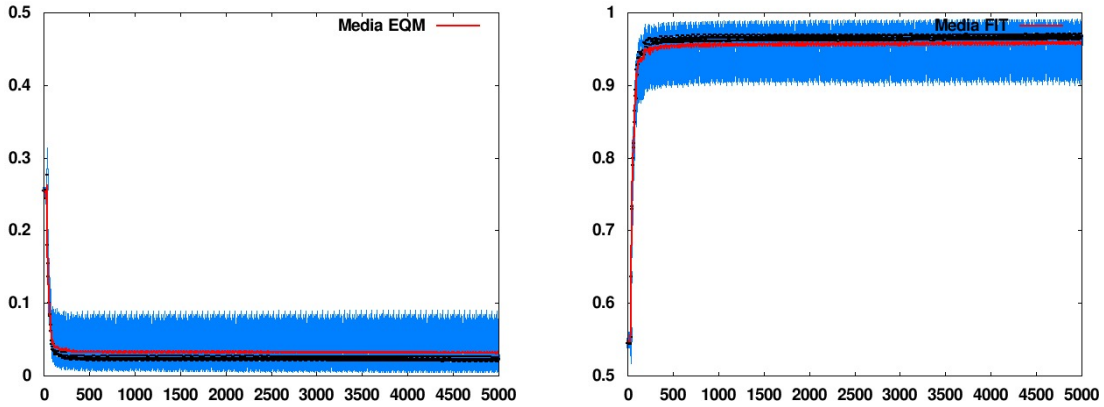


(b) Imagem 13 do Cenário 2



(c) Imagem 14 do Cenário 2

Figura 4.10: Experimento no Cenário 2. Comparação dos resultados entre os ciclos 500 e 1000 da mesma RNA.



(a) EQM - Taxa de erro

(b) FIT - Taxa de acerto

Figura 4.11: Comportamento semelhante dos resultados EQM e FIT.

4.3 Sistema de Identificação

Para avaliar o sistema de identificação proposto, foi realizado um outro experimento com os conjuntos de atributos de imagem escolhidos na etapa de seleção de atributos. Com base nos três conjuntos de atributos selecionados foram compostas seis RNA diferentes - duas instâncias para cada conjunto. Desta forma, o sistema de identificação calculará a média dos resultados das RNA para compor uma única classificação. Os conjuntos de atributos são:

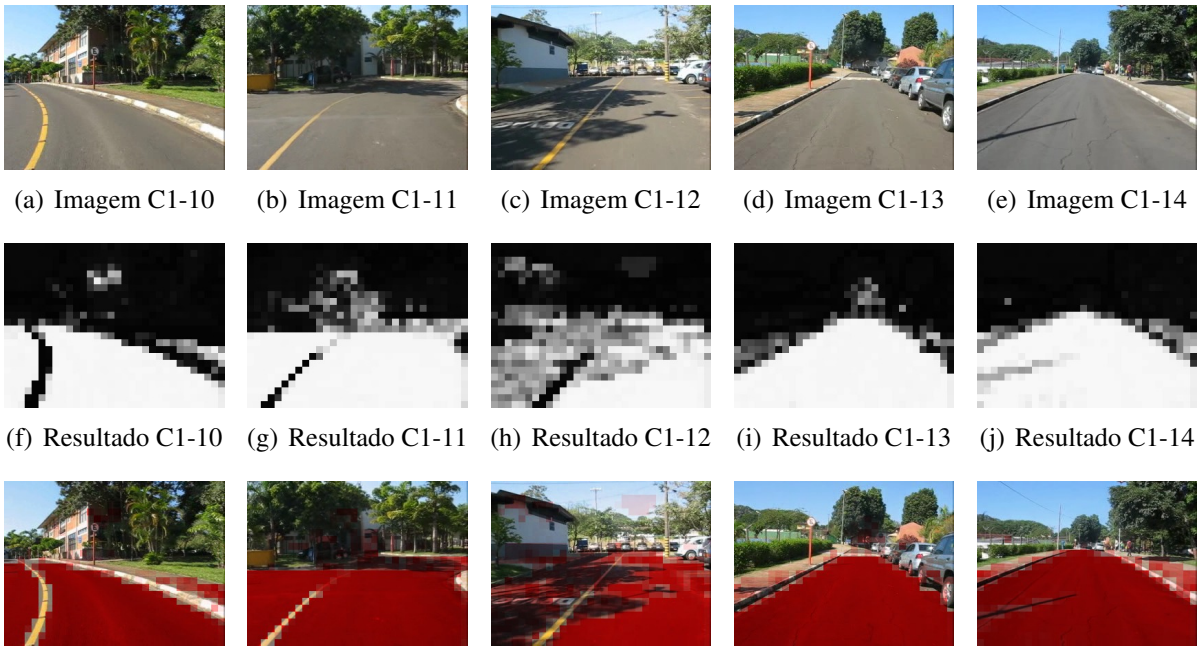
- Média U, Média V, Média B Normalizado, Entropia H, Energia G Normalizado e Média H
- Média V, Entropia H, Energia G Normalizado, Média G, Média U, Média R, Média H, Média B Normalizado, Média G Normalizado e Entropia Y.
- Média U, Média B Normalizado, Média V, Variância B, Média S, Média H, Média G Normalizado, Entropia G Normalizado.

Assim como no experimento de seleção de atributos, foi gerado um conjunto de padrões para cada um dos cenários do Apêndice C e um outro conjunto que contem padrões de todos eles, este último conjunto foi chamado de Cenário 9 (C9). Treinar o sistema de identificação significa treinar cada uma das seis RNA que o compõe. A etapa de treinamento de cada uma dessas topologias de RNA consiste em instanciar cinco vezes a mesma topologia e treiná-las até 5000 ciclos. Foi constatado por métodos empíricos que este número de ciclos é suficiente para a generalização das RNA. Após o término do treinamento destas cinco instâncias, seleciona-se

aquela que obteve o melhor desempenho de classificação (FIT), para fazer parte do sistema de identificação. Ao final do treinamento do sistema, seis RNA estão prontas para realizar a tarefa de classificação de elementos de imagem.

Modelo de Treinamento	Conjuntos de Validação								
	C1	C2	C3	C4	C5	C6	C7	C8	C9
Cenário 1	0,94	0,89	0,89	0,89	0,92	0,92	0,89	0,90	0,91
Cenário 2	0,99	0,97	0,84	0,96	0,88	0,96	0,93	0,89	0,92
Cenário 3	0,92	0,87	0,89	0,87	0,90	0,89	0,89	0,89	0,89
Cenário 4	0,84	0,94	0,79	0,96	0,84	0,94	0,91	0,87	0,89
Cenário 5	0,91	0,86	0,87	0,87	0,92	0,89	0,88	0,87	0,88
Cenário 6	0,84	0,94	0,82	0,95	0,84	0,95	0,90	0,86	0,89
Cenário 7	0,75	0,73	0,68	0,72	0,78	0,77	0,96	0,93	0,77
Cenário 8	0,85	0,76	0,79	0,76	0,85	0,82	0,95	0,95	0,83
Cenário 9	0,92	0,93	0,87	0,93	0,91	0,93	0,91	0,90	0,91

Tabela 4.5: Resultados do sistema para todos os cenários apresentados. Cada coluna representa o FIT obtido em padrões de um cenário. Cada linha mostra a origem dos padrões com que o sistema foi treinado.



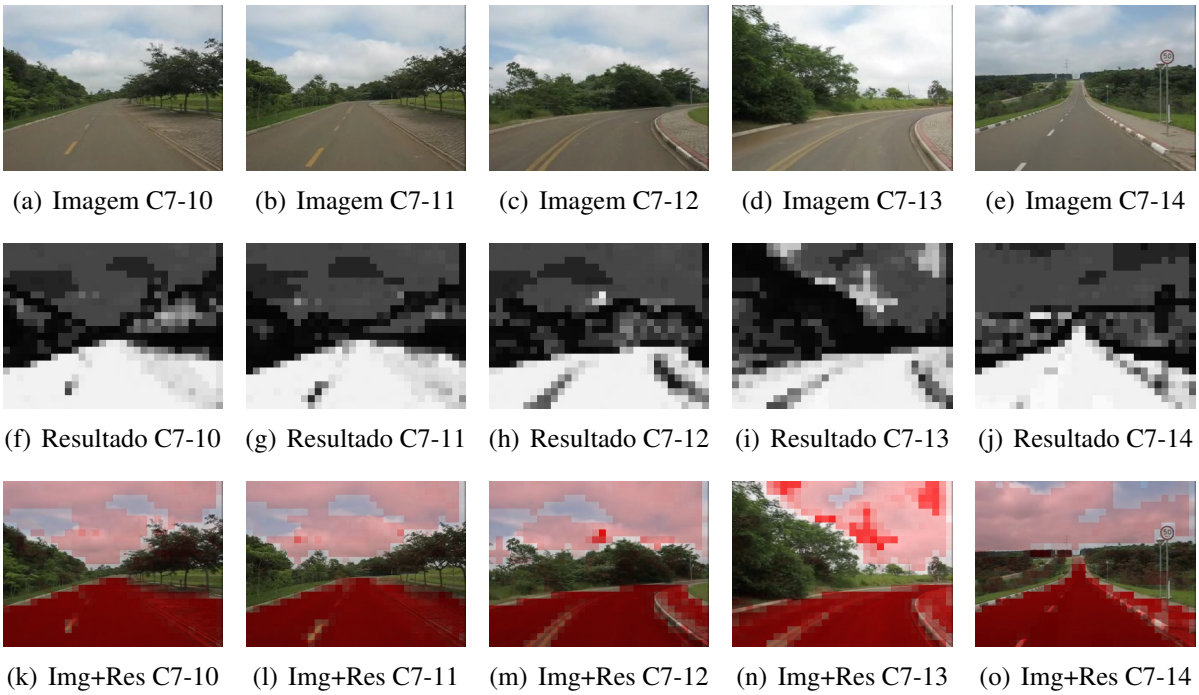
(a) Imagem C1-10 (b) Imagem C1-11 (c) Imagem C1-12 (d) Imagem C1-13 (e) Imagem C1-14

(f) Resultado C1-10 (g) Resultado C1-11 (h) Resultado C1-12 (i) Resultado C1-13 (j) Resultado C1-14

(k) Img+Res C1-10 (l) Img+Res C1-11 (m) Img+Res C1-12 (n) Img+Res C1-13 (o) Img+Res C1-14

Figura 4.12: Exemplo de resultado bom: Resultados sobre o Cenário 1 do sistema treinado também com padrões do Cenário 1. O valor do FIT é $0,94$. As imagens (a)-(e) mostram as imagens originais. As imagens (f)-(j) mostram o resultado das classificações. As imagens (k)-(o) mostram o resultado sobre a imagem real, onde quanto mais certeza sobre a navegabilidade, mais vermelho é o bloco. Os blocos considerados não navegáveis mantém sua cor original.

O sistema de identificação foi treinado diversas vezes, onde cada treinamento foi realizado usando um dos conjuntos de padrões criados. Para cada treinamento, o desempenho (FIT) do sistema de identificação foi calculado para todos os outros conjuntos de padrões. Essa avaliação tem como objetivo verificar a robustez do sistema, ou seja, a robustez da combinação das várias classificações. O FIT de cada um dos treinamentos para cada um dos cenários é mostrado na Tabela 4.5. A linha "Cenário 1" representa o sistema treinado com padrões do Cenário 1, pode-se ver que seu desempenho geral foi muito bom uma vez que o menor FIT alcançado foi $0,89$. Assim como a primeira linha, todos os outros resultados também mostraram um bom desempenho geral exceto quando o sistema foi treinado pelos cenários 7 e 8. A principal prova do bom desempenho geral pode ser concluída a partir da última coluna que é a avaliação do sistema para padrões pertencentes a todos os cenários. Nesta última coluna o sistema conseguiu bons resultados em sete dos nove treinamentos. Outra prova é mostrada na última linha que é quando o sistema é treinado com padrões de todos os cenários, ficando com bom desempenho de classificação em todos os casos.



(a) Imagem C7-10 (b) Imagem C7-11 (c) Imagem C7-12 (d) Imagem C7-13 (e) Imagem C7-14

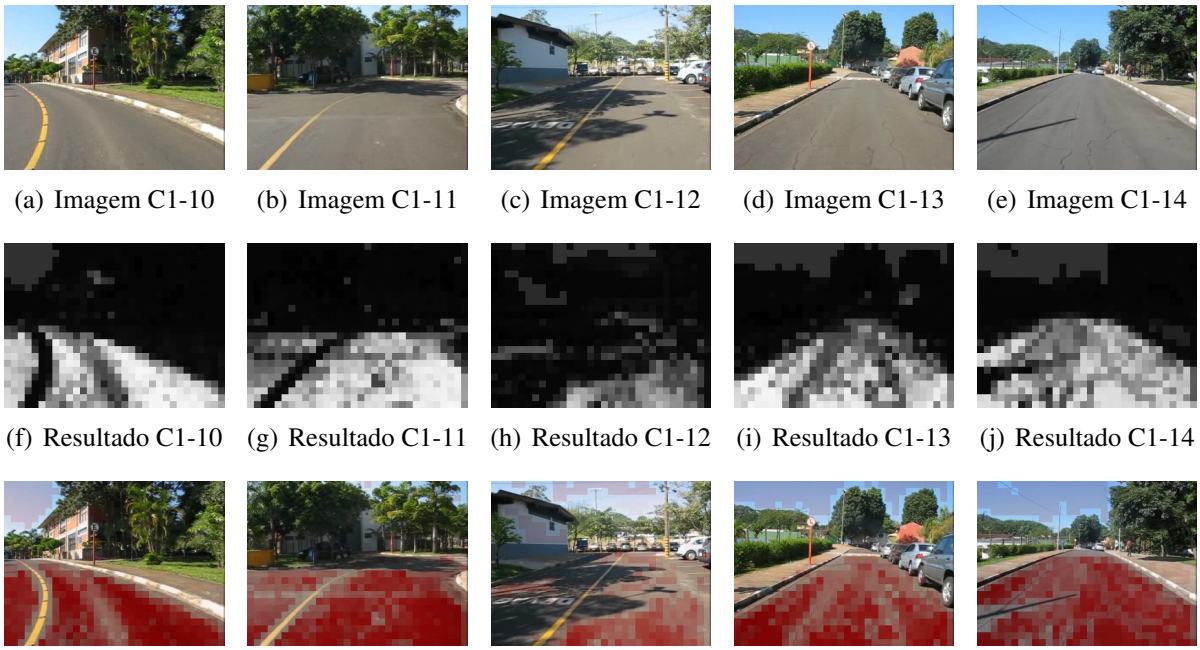
(f) Resultado C7-10 (g) Resultado C7-11 (h) Resultado C7-12 (i) Resultado C7-13 (j) Resultado C7-14

(k) Img+Res C7-10 (l) Img+Res C7-11 (m) Img+Res C7-12 (n) Img+Res C7-13 (o) Img+Res C7-14

Figura 4.13: Exemplo de resultado aceitável: Resultados sobre o Cenário 7 do sistema treinado com padrões do Cenário 5. O valor do FIT é $0,88$. As imagens (a)-(e) mostram as imagens originais. As imagens (f)-(j) mostram o resultado das classificações. As imagens (k)-(o) mostram o resultado sobre a imagem real, onde quanto mais certeza sobre a navegabilidade, mais vermelho é o bloco. Os blocos considerados não navegáveis mantém sua cor original.

Constatou-se que um FIT maior que $0,9$ significa um bom desempenho e um FIT maior que $0,85$ ainda é um desempenho aceitável. Outro valor importante é que um FIT menor que $0,8$ não resulta em uma boa classificação. Tais afirmações foram obtidas por meio da análise dos resultados deste experimento. A Figura 4.12 mostra resultados sobre o Cenário 1 do sistema também treinado para o Cenário 1 para exemplificar quando o sistema tem bons desempenhos. As imagens (a)-(e) mostram as imagens originais. As imagens (f)-(j) mostram o resultado das classificações. As imagens (k)-(o) mostram o resultado sobre a imagem real, onde quanto mais certeza sobre a navegabilidade, mais vermelho é o bloco. Os blocos considerados não navegáveis mantém sua cor original.

Visualizando as imagens desse resultado pode-se notar que o grau de certeza desse sistema é elevado mesmo em condições extremamente conflitantes como a presença de uma forte sombra na imagem C1-12. Além disso seu grau de certeza sobre a não-navegabilidade do céu também é alto. Outro desempenho considerado bom, mas marcado como aceitável devido ao seu FIT, foi o treinamento do sistema que alcançou o FIT $0,88$ com treinamento no Cenário 5 e avaliado com os padrões do Cenário 7, como mostra a Figura 4.13. O que fez a nota desse sistema abaixar foi a classificação errada do céu, pois apesar da incerteza ser pequena, o número de blocos pertencentes ao céu é alto. Todos resultados com FIT semelhantes foram muito parecidos visualmente.



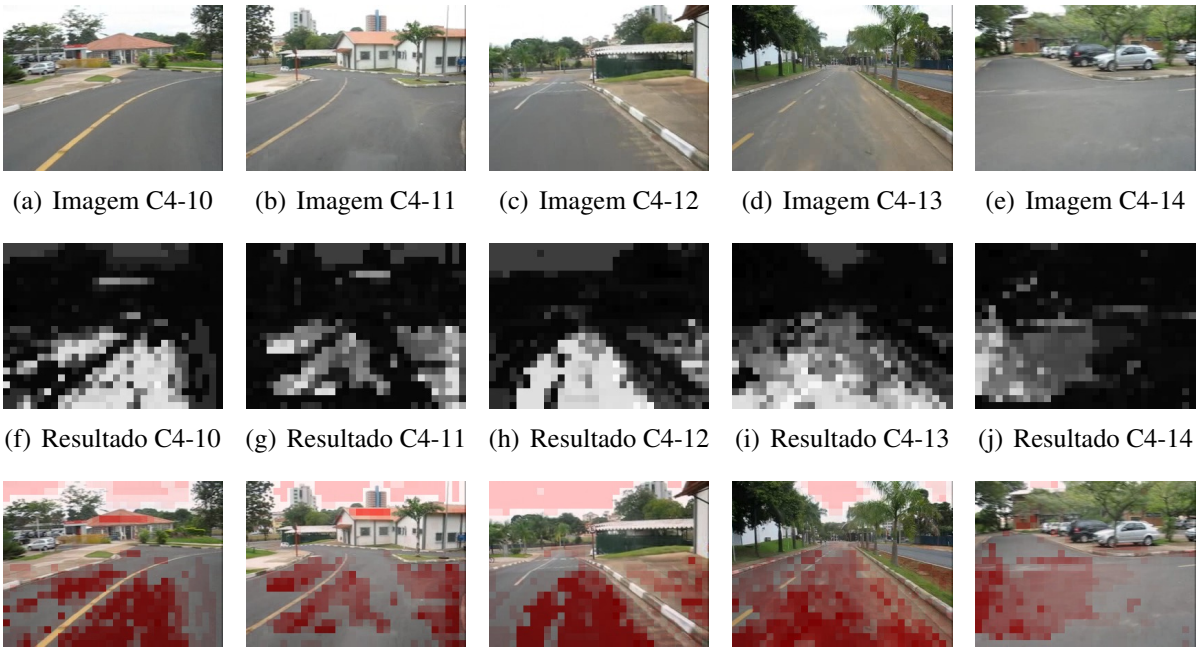
(a) Imagem C1-10 (b) Imagem C1-11 (c) Imagem C1-12 (d) Imagem C1-13 (e) Imagem C1-14

(f) Resultado C1-10 (g) Resultado C1-11 (h) Resultado C1-12 (i) Resultado C1-13 (j) Resultado C1-14

(k) Img+Res C1-10 (l) Img+Res C1-11 (m) Img+Res C1-12 (n) Img+Res C1-13 (o) Img+Res C1-14

Figura 4.14: Resultados sobre o Cenário 1 do sistema treinado com padrões do Cenário 7. O valor do FIT é $0,75$.

Resultados relativamente ruins também foram exemplificados a fim de expor o motivo de determinar o uso do sistema de identificação apenas se este conseguir alcançar um FIT maior ou igual a $0,85$. As Figuras 4.14 e 4.15 mostram resultados com pouco grau de certeza, em alguns casos deixando muito a desejar. A Figura 4.14 mostra resultados do sistema treinado para o Cenário 7 classificando imagens do Cenário 1, as imagens mostram claramente que o sistema não consegue classificar corretamente áreas afetadas pela sombra e também comete pequenos erros no céu. Os resultados da Figura 4.15 mostram que o mesmo sistema que não conseguiu classificar a rua com sombra corretamente também não conseguiu classificar a rua em condições de iluminação constante e o céu nublado. Baseado nessas observações, o sistema treinado com os padrões do Cenário 7 não poderia ser utilizado em cenários complexos. Em outras palavras, não se pode garantir que o sistema consiga manter uma boa classificação se o percurso sofrer significativas alterações de iluminação e cor, uma vez que o sistema não recebeu um conjunto de padrões de treinamento completo.



(a) Imagem C4-10 (b) Imagem C4-11 (c) Imagem C4-12 (d) Imagem C4-13 (e) Imagem C4-14

(f) Resultado C4-10 (g) Resultado C4-11 (h) Resultado C4-12 (i) Resultado C4-13 (j) Resultado C4-14

(k) Img+Res C4-10 (l) Img+Res C4-11 (m) Img+Res C4-12 (n) Img+Res C4-13 (o) Img+Res C4-14

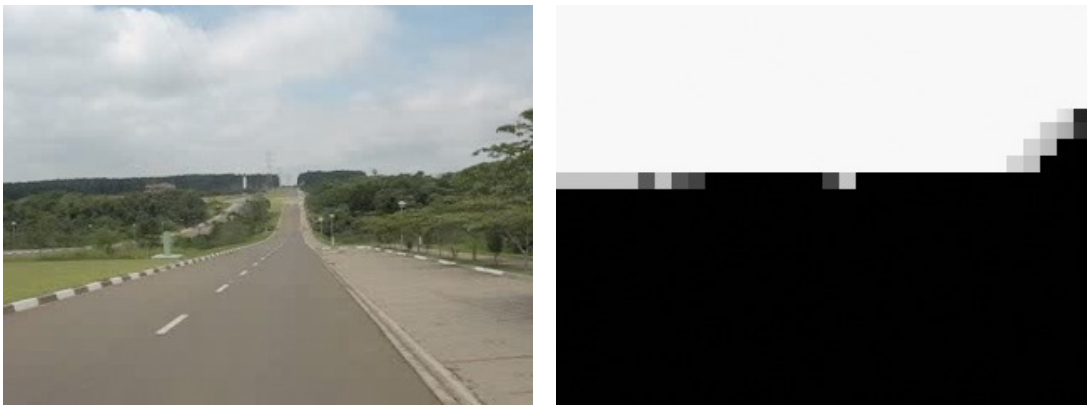
Figura 4.15: Resultados sobre o Cenário 4 do sistema treinado com padrões do Cenário 7. O valor do FIT é $0,72$. As imagens (k)-(o) mostram o resultado sobre a imagem real, onde quanto mais certeza sobre a navegabilidade, mais vermelho é o bloco.

Quando o sistema de identificação não alcança o desempenho desejado, o usuário deve recriar o conjunto de treinamento de forma que este se torne mais amplo possuindo mais exemplos de todas ocorrências de cores incluir imagens com sujeira, sombras e outros. Esse caso pôde ser muito bem exemplificado quando o sistema foi treinado com os padrões do Cenário 9, pois este

sistema obteve bons desempenhos de classificação em todos os cenários. Para o sistema em uso, não é conveniente que ele treine 5 vezes a mesma RNA e utilize a com melhor desempenho, pois assim o sistema com 6 RNA demoraria em torno de 40 minutos para todas as RNA convergirem em um Computador Core 2 Duo 2.0GHZ. Desta forma, baseado nos resultados descritos nesta seção, deve-se determinar um FIT mínimo em torno de $0,9$ para as RNA (em cenários simples pode-se elevar esse mínimo), assim não é necessário completar os 5 treinamentos de cada RNA. De acordo com os experimentos, o treinamento do sistema em locais parecidos com os cenários 7 e 8 deve ser muito rápido.

4.4 Identificação da Linha do Horizonte

Outro trabalho derivado dessa dissertação é a detecção da linha do horizonte em imagens obtidas pelo veículo. Esta seção explica resumidamente um experimento realizado sobre alguns vídeos a fim de demonstrar outra aplicação do sistema proposto. A motivação desse experimento é que na maioria das vezes, quase metade da imagem é composta pelo céu. A região de interesse para a classificação é apenas a região abaixo da linha do horizonte, pois é a região da imagem que representa o terreno. Detectando a linha do horizonte pode-se reduzir o tempo de processamento uma vez que praticamente metade da imagem não será processada. Constatou-se também que a RNA necessária para diferenciar o céu é muito mais simples que a utilizada na identificação de superfícies navegáveis.



(a) Imagem Original (b) Resultado da Classificação

Figura 4.16: Selecionando região de interesse do sistema de identificação.

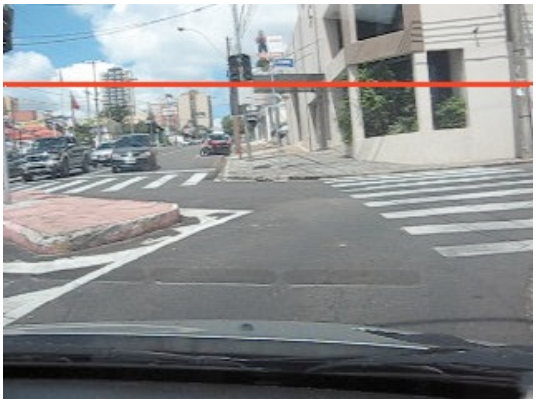
Assim como na identificação de superfícies navegáveis, neste experimento foram selecionados blocos que representam chão e outros que representam o céu para o treinamento do sistema.

Após o treinamento realiza-se a classificação. A Figura 4.16 mostra a imagem original e o resultado da classificação. Após essa classificação pode-se estimar a altura da linha do horizonte e utilizar o sistema de identificação apenas nos blocos abaixo dessa linha. A Figura 4.17 mostram alguns resultados em cenários considerados simples (onde a diferenciação de cores é bem nítida e não existe a oclusão do céu) e cenários considerados complexos (presença de inúmeras cores e oclusão do céu por prédios e casas).

(a) Cena 1

(b) Cena 2

(c) Cena 3



(d) Cena 4

Figura 4.17: Resultado mostrando a estimativa da linha do horizonte na imagem em quatro cenários distintos.

Como o objetivo era melhorar o desempenho, a detecção da linha do horizonte foi executada em uma frequência de 1Hz. Este experimento foi executado em um computador Pentium Dual Core 2.16GHz com 2GB de memória RAM e em um Core i5 operando a 2.66GHz com 4GB de memória, ambos com o sistema operacional Ubuntu 9.04. Todo o experimento foi realizado em modo *"off-line"* sobre vídeos coletados em diferentes trajetos, e o cálculo de *"frames por segundo"* (FPS) foi obtido de acordo com a média do tempo de execução da classificação para

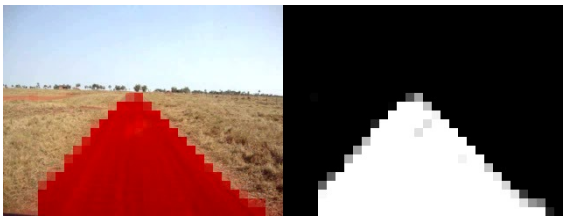
cada imagem. Os resultados do desempenho aproximado são mostrados na Tabela 4.6. Pode-se perceber um aumento de performance quando se utiliza a detecção da linha do horizonte. É importante ressaltar que o sistema utilizado foi mais simples que o modelo final desta dissertação. O sistema foi composto de apenas 4 RNA com no máximo 5 atributos de entrada cada uma. Todo esse experimento foi realizado com o objetivo de apenas exemplificar uma outra aplicação de uso do sistema proposto.

Processador	Sem Detecção do Horizonte	Com Detecção do Horizonte
Dual Core 2.16GHz	34 FPS	54 FPS
Core i5 2.66GHz	95 FPS	120 FPS

Tabela 4.6: Comparação do desempenho do sistema utilizando detecção do horizonte ou não.

Baseado nos resultados desse experimento pode-se concluir que detectar a linha do horizonte faz o sistema ficar aproximadamente 60% mais rápido no Dual core e 25% no Core i5.

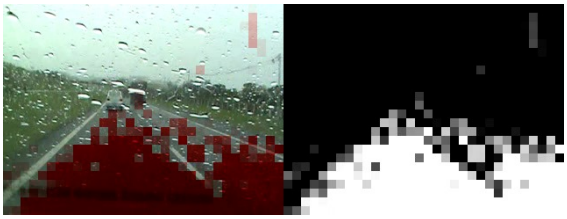
4.5 Outros Resultados



(a) Estrada de terra

(b) Rodovia durante a noite

(c) Rodovia durante o dia



(d) Rodovia durante uma chuva

Figura 4.18: Experimentos em diferentes cenários.

Com o objetivo de demonstrar a robustez do sistema desenvolvido , foram realizadas outras avaliações em cenários diferentes daqueles usados nos experimentos apresentados nas seções anteriores . Ou seja, cenários que não foram usados na etapa de seleção de atributos nem na etapa de avaliação do sistema. O algoritmo foi treinado com alguns padrões das primeiras imagens dos vídeos e os resultados são mostrados nas figuras 4.18, onde as imagens da esquerda

apresentam o resultado da classificação sobre as imagens originais. As regiões consideradas navegáveis são coloridas em vermelho e as regiões não navegáveis mantém sua cor original. A imagem da direita mostra o grau de certeza que o sistema tem sobre a navegabilidade dos blocos, onde a cor branca indica maior índice de navegabilidade.

Um dos casos de teste é um ambiente com uma estrada de terra. O resultado é mostrado na Figura 4.18(a). Outro caso é uma rodovia durante a noite mostrado na Figura 4.18(b). Outra rodovia, mas durante o dia, é mostrada na Figura 4.18(c). E finalmente, uma imagem de uma rodovia com chuva Figura 4.18(d). Baseado em uma análise visual, todos os resultados da classificação foram bastante satisfatórios.

4.6 Considerações

Baseado nos resultados dos vários experimentos deste capítulo, vários parâmetros de configuração do sistema de identificação de superfícies navegáveis puderam ser determinados. Os atributos de imagem usados na classificação foram escolhidos baseado em vários experimentos. A métrica de avaliação de RNA foi escolhida com o objetivo de selecionar a melhor RNA de forma mais adequada. E parâmetros foram definidos para compor a fase de treinamento do sistema, como o número máximo de ciclos de treinamento de cada RNA, o desempenho de classificação mínimo que todo o sistema deve possuir antes de ser utilizado. Além disso, foi realizado um estudo sobre o tempo de treinamento do sistema. A partir dessas várias conclusões foram definidas algumas condições que se satisfeitas encerram o treinamento do sistema, pois significam que o sistema já pode ser usado.

As configurações finais do sistema de identificação de superfícies navegáveis são:

- Utilizar 6 RNA: 2 instâncias de cada topologia, onde a camada de entrada foi definida pelos atributos de imagem utilizados, a camada intermediária e de saída são constantes como a metodologia definiu.
- Utilizar a métrica de avaliação de RNA chamada FIT.
- Treinar cada RNA até completar 5000 ciclos ou seu FIT ultrapassar a nota $0,9$. Se o FIT não alcançar esse desempenho em 5000 ciclos são realizadas mais 5 tentativas.
- O supervisor deve redefinir o conjunto de treinamento de forma que este seja mais completo que o anterior se ao final do treinamento de todas as RNA, o sistema possuir um FIT menor que $0,85$.

CAPÍTULO

5

Conclusão

Esta dissertação apresenta um sistema de identificação de superfícies navegáveis baseado em visão computacional e redes neurais artificiais. O resultado dessa identificação é um mapa visual de navegabilidade que possui graus de certeza da navegabilidade para cada elemento criado a partir da imagem original. Esse mapa pode ser usado por sistemas de tomada de decisão para controle de um robô.

Para o desenvolvimento deste trabalho, foi realizado um estudo sobre diferentes abordagens de processamento e diferentes características de imagem a fim de escolher uma forma de se processar uma sequência de imagens obtida por uma câmera acoplada em um veículo. Posteriormente, o método de aprendizado de máquina conhecido por redes neurais artificiais também foi estudado e implementado para realizar a tarefa de classificar os elementos de imagem entre navegáveis e não-navegáveis. Vale ressaltar que foi implementado um método de avaliação de RNA que leva em consideração a precisão do erro para cada padrão apresentado a RNA. O modelo final do classificador possui várias redes neurais artificiais que utilizam diferentes atributos de imagem a fim de dar maior robustez ao sistema de identificação.

Foi possível observar que combinar diferentes atributos de cores de diferentes espaços de cores eleva a capacidade de classificação da RNA em diferentes condições de iluminação ou ambiente. Nela também foram determinados alguns parâmetros muito importantes que devem ser considerados se este sistema for utilizado em sistemas de tempo real. Uma das contribuições deste projeto foi o desenvolvimento de um método de avaliação que combina as informações

do EQM e do *HIT RATE*. Pois para o problema da identificação de superfícies navegáveis, é importante saber o quão imprecisa está a classificação do conjunto de padrões referente a uma imagem. Outra contribuição relevante foi o estudo dos atributos de imagem que possuem melhor desempenho de classificação de imagens de ambientes externos. Esses atributos de cor e textura poderão ser utilizados em futuros métodos de classificação.

Além disso, este trabalho contribuiu para o projeto do Laboratório de Robótica Móvel do ICMC-USP em parceria com o INCT-SEC e o projeto SENA que tem como proposta o desenvolvimento de um veículo autônomo. Uma vez que este sistema de identificação poderá fazer parte do sistema de navegação desses veículos autônomo.

5.1 Trabalhos Futuros

Como trabalho futuro, este sistema pode ser utilizado em conjunto com um sistema baseado em sensores de distância a fim de aumentar sua robustez e desempenho, além de permitir uma adaptabilidade não supervisionada. Essa integração permitirá ao sistema a capacidade de considerar informações como altura de uma região, em vez de apenas cores e texturas, para classificar as regiões entre navegáveis e não-navegáveis.

Na mesma linha de pesquisa em visão computacional, este sistema pode ser ampliado para detectar faixas de trânsito que de acordo com os experimentos realizados neste trabalho interferem no desempenho do classificador. Além disso, é possível implementar métodos de processamento de imagem que eliminam ou diminuem a influência de sombras na imagem antes de passar pela etapa de geração de elementos de imagem. Outros métodos que ajudarão na eficiência desse sistema é a detecção de carros, pessoas e sinais de trânsito como placas e faixas de pedestres.

Como este método classifica porções de imagem de forma independente ele também pode ser paralelizado visando maior rapidez no sistema. Recomenda-se uso de GPU ou FPGA para estes sistemas embarcados uma vez que não existe espaço físico para grandes clusters dentro de um robô móvel. Além da paralelização, pode-se utilizar programação concorrente. Todos essas possibilidades de implementação tem como objetivo analisar e melhorar a velocidade de processamento do sistema.

Referências Bibliográficas

ANTP http://noticias.uol.com.br/economia/ultnot/valor/2007/09/18/ult1913u76072.jhtm, visitado em Janeiro, 2010.

BARANAUSKAS, J. A.; MONARD, M. C. Combining symbolic classifiers from multiple inducers. *Knowledge-Based Systems*, v. 16, n. 3, p. 129 – 136, 2003.

BONIN-FONT, F.; ORTIZ, A.; OLIVER, G. Visual navigation for mobile robots: A survey. *Journal of Intelligent & Robotic Systems*, v. 53, p. 263–296, 10.1007/s10846-008-9235-4, 2008.
Disponível em http://dx.doi.org/10.1007/s10846-008-9235-4

BRADSKI, G.; KAEHLER, A. *Learning opencv: Computer vision with the opencv library*. Cambridge, MA: O'Reilly, 2008.

BRAGA, A. P.; DE CARVALHO, A. P. L.; LUDERMIR, T. B. *Redes neurais artificiais - teoria e aplicações*. 2 ed. Rio de Janeiro: LTC, 2000.

CARBONELL, J. G.; MICHALSKI, R. S.; MITCHELL, T. M. An overview of machine learning. In: MICHALSKI, R. S.; CARBONELL, J. G.; MITCHELL, T. M., eds. *Machine Learning: An Artificial Intelligence Approach*, Berlin, Heidelberg: Springer, p. 3–23, 1984.

CARNEGIEMELLON, U. http://www-2.cs.cmu.edu/afs/cs/project/chopper/www/, visitado em Setembro, 2010.

CLARK, M.; BOVIK, A. C.; GEISLER, W. S. Texture segmentation using gabor modulation/-demodulation. *Pattern Recogn. Lett.*, v. 6, n. 4, p. 261–267, 1987.

COSTA, L. D. F. D.; CESAR, JR., R. M. *Shape analysis and classification: Theory and practice*. Boca Raton, FL, USA: CRC Press, Inc., 2000.

COULEUR http://www.couleur.org, visitado em Setembro, 2010.

CRISMAN, J.; THORPE, C. Unscarf, a color vision system for the detection of unstructured roads. In: *Proceedings of IEEE International Conference on Robotics and Automation*, 1991, p. 2496 – 2501.

CRISMAN, J.; THORPE, C. Scarf: a color vision system that tracks roads and intersections. *Robotics and Automation, IEEE Transactions on*, v. 9, n. 1, p. 49 –58, 1993.

CRISMAN, J. D.; THORPE, C. E. Color vision for road following. In: *Vision and Navigation: The CMU Navlab, C. Thorpe (Ed*, Kluwer Academic Publishers, 1988, p. 9–24.

DARPA http://www.darpa.mil/, visitado em Março, 2009.

FANN http://leenissen.dk/fann/html/files/fann-h.html, visitado em Setembro, 2010.

FOEDISCH, M. Adaptive real-time road detection using neural networks. In: *in Proc. 7th Int. Conf. on Intelligent Transportation Systems, Washington D.C*, 2004, p. 167 – 172.

FOEDISCH, M.; TAKEUCHI, A. Adaptive road detection through continuous environment learning. In: *Applied Imagery Pattern Recognition Workshop*, 2004, p. 13–15.

GOOL, L. J. V.; DEWAELE, P.; OOSTERLINCK, A. Texture analysis anno 1983. *Computer Vision, Graphics, and Image Processing*, v. 29, n. 3, p. 336–357, 1985.

GRUDIC, G. Z.; MULLIGAN, J. Outdoor path labeling using polynomial mahalanobis distance. In: *Robotics: Science and Systems*, 2006, p. 4.

HARALICK, R. M. Statistical and structural approaches to texture. *Proceedings of the IEEE*, v. 67, n. 5, p. 786–804, 1979.

HAYKIN, S. *Neural networks: A comprehensive foundation (2nd edition).* 2 ed. Prentice Hall, 1998.

IBGE http://www.ibge.gov.br/home/, visitado em Janeiro, 2009.

ITARDA http://www.itarda.or.jp. Visitado em Fevereiro, 2010.

JOBLOVE, G. H.; GREENBERG, D. Color spaces for computer graphics. *SIGGRAPH Comput. Graph.*, v. 12, n. 3, p. 20–25, 1978.

JUNG, C.; KELBER, C. A lane departure warning system using lateral offset with uncalibrated camera. In: *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, 2005, p. 102 – 107.

KHALIFA, O.; ASSIDIQ, A.; HASHIM, A.-H. Vision-based lane detection for autonomous artificial intelligent vehicles. In: *Semantic Computing, 2009. ICSC '09. IEEE International Conference on*, 2009, p. 636–641.

LEE, J.; III, C. D. C.; KIM, S.; KIM, J., eds. *Road following in an unstructured desert environment using monocular color vision as applied to the darpa grand challenge*, International Conference on Control, Automation and Systems, 2005.

MITCHELL, T. *Machine learning (mcgraw-hill international edit)*. 1st ed. McGraw-Hill Education (ISE Editions), 1997.

NASA `http://mpfwww.jpl.nasa.gov/default.html`, visitado em Março, 2009.

POMERLEAU, D. Neural network vision for robot driving. In: ARBIB, M., ed. *The Handbook of Brain Theory and Neural Networks*, p. 161–181, 1996.

RASMUSSEN, C. Combining laser range, color, and texture cues for autonomous road following. In: *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, 2002, p. 4320 – 4325 vol.4.

REITER, C. With j: image processing 2: color spaces. *SIGAPL APL Quote Quad*, v. 34, n. 3, p. 3–12, 2004.

RIEDMILLER, M.; BRAUN, H. *Rprop – description and implementation details*. Relatório Técnico, Universitat Karlsruhe, 1994.
Disponível em `citeseer.ist.psu.edu/riedmiller94rprop.html`

ROJAS, R. *Neural networks: a systematic introduction*. New York, NY, USA: Springer-Verlag New York, Inc., 1996.

RONCATTI, M. A. *Avaliação de métodos Ótimos e subótimos de seleção de características de texturas em imagens*. Dissertação de Mestrado, ICMC - Universidade de São Paulo, 2008.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. *Learning internal representations by error propagation* Cambridge, MA, USA: MIT Press, p. 318–362, 1986.
Disponível em `http://portal.acm.org/citation.cfm?id=104279.104293`

RUSSELL, S. J.; NORVIG, P. *Artificial intelligence: A modern approach*. Pearson Education, 2003.
Disponível em `http://portal.acm.org/citation.cfm?id=773294`

SALES, D.; SHINZATO, P. Y.; ; PESSIN, G.; OSÓRIO, F.; WOLF, D. F. Autonomous vision-based navigation using fsm control. In: *Latin American Robotics Symposium*, IEEE, 2010.

DOS SANTOS, D. P. *Seleção de caracteríticas: abordagem via redes neurais aplicada à segmentação de imagens*. Dissertação de Mestrado, ICMC - Universidade de São Paulo, 2007.

SHANNON, C. E. A mathematical theory of communication. *Bell system technical journal*, v. 27, 1948.

SHINZATO, P.; FERNANDES, L.; OSORIO, F.; WOLF, D. Path recognition for outdoor navigation using artificial neural networks: Case study. In: *Industrial Technology (ICIT), 2010 IEEE International Conference on*, 2010, p. 1457 –1462.

SHINZATO, P.; WOLF, D. A road following approach using artificial neural networks combinations. *Journal of Intelligent & Robotic Systems*, p. 1–20, 10.1007/s10846-010-9463-2, 2010a.
Disponível em `http://dx.doi.org/10.1007/s10846-010-9463-2`

SHINZATO, P. Y.; WOLF, D. F. Statistical analysis of image-features used as inputs of an road identifier based in artificial neural networks. In: *Latin American Robotics Symposium*, IEEE, 2010b.

SOUZA, J.; SALES, D.; SHINZATO, P. Y.; OSÓRIO, F.; WOLF, D. F. Template-based autonomous navigation in urban environments. In: *Accepted in 26th SAC: Proceedings of the 2011 ACM Symposium on Applied Computing*, TaiChung, Taiwan: ACM, 2011, p. 1–6.

STANFORD, U. `http://heli.stanford.edu/`, visitado em Setembro, 2010.

TAN, C.; HONG, T.; CHANG, T.; SHNEIER, M., eds. *Color model-based real-time learning for road following*, IEEE Intelligent Transportation Systems Conference, 2006.

THORPE, C.; HEBERT, M.; KANADE, T.; SHAFER, S. *Vision and navigation for the Carnegie-Mellon Navlab*, v. 10, n. 1, p. 362 – 373, 1988.

THRUN, S.; MONTEMERLO, M.; DAHLKAMP, H.; STAVENS, D.; ARON, A.; DIEBEL, J.; FONG, P.; GALE, J.; HALPENNY, M.; HOFFMANN, G.; LAU, K.; OAKLEY, C.; PALATUCCI, M.; PRATT, V.; STANG, P.; STROHBAND, S.; DUPONT, C.; JENDROSSEK, L.-E.; KOELEN, C.; MARKEY, C.; RUMMEL, C.; VAN NIEKERK, J.; JENSEN, E.; ALESSANDRINI, P.; BRADSKI, G.; DAVIES, B.; ETTINGER, S.; KAEHLER, A.; NEFIAN, A.; MAHONEY, P. Winning the darpa grand challenge. *Journal of Field Robotics*, 2006.

THRUN, S.; THAYER, S.; WHITTAKER, W.; BAKER, C.; BURGARD, W.; FERGUSON, D.; HÄHNEL, D.; MONTEMERLO, M.; MORRIS, A.; OMOHUNDRO, Z.; REVERTE, C.; WHITTAKER, W. Autonomous exploration and mapping of abandoned mines. *IEEE Robotics and Automation Magazine*, p. 79–91, 2005.

WILLIAMS, S. B.; MAHON, I. Design of an unmanned underwater vehicle for reef surveying. In: *IFAC 3rd IFAC Symposium on Mechatronic Systems*, 2004.

APÊNDICE

A

Publicações Obtidas Como Resultado Desse Trabalho

1. SHINZATO, P. Y.; WOLF, D. F. A road following approach using artificial neural networks combinations. *Journal of Intelligent & Robotic Systems*, p. 1–20, 10.1007/s10846-010-9463-2, 2010a.
   Disponível em `http://dx.doi.org/10.1007/s10846-010-9463-2`

2. SHINZATO, P. Y.; WOLF, D. F. Path recognition for outdoor navigation. In: *Latin American Robotics Symposium*, IEEE, 2009.

3. SHINZATO, P. Y.; FERNANDES, L. C.; OSÓRIO, F. S.; WOLF, D. F. Path recognition for outdoor navigation using artificial neural networks: Case study. In: *International Conference on Industrial Technology*, IEEE, 2010.

4. SHINZATO, P. Y.; WOLF, D. F. Features Image Analysis for Road Following Algorithm Using Neural Networks. In: *7th IFAC Symposium on Intelligent Autonomous Vehicles*, 2010b, Lecce - Itália.

5. SHINZATO, P. Y.; WOLF, D. F. Statistical analysis of image-features used as inputs of an road identifier based in artificial neural networks. In: *Latin American Robotics Symposium*, IEEE, 2010c.

6. SALES, D. O.; SHINZATO, P. Y.; PESSIN, G.; OSÓRIO, F. S.; WOLF, D. F. Vision-Based Autonomous Navigation System Using ANN and FSM Control. In: *Latin American Robotics Symposium*, IEEE, 2010.

7. SOUZA, J.; SALES, D. O.; SHINZATO, P. Y.; OSÓRIO, F. S.; WOLF, D. F. Template-based autonomous navigation in urban environments. Accepted In: *26th SAC: Proceedings of the 2011 ACM Symposium on Applied Computing*, ACM, 2011.

8. HATA, A. Y.; SHINZATO, P. Y.; WOLF, D. F. Mapeamento e Classificação de Terrenos Utilizando Aprendizado Supervisionado.. In: *Congresso Brasileiro de Automática*, 2010.

APÊNDICE

B

Lista de Atributos Avaliados

- Média do canal de cor R do espaço de cor RGB
- Média do canal de cor G do espaço de cor RGB
- Média do canal de cor B do espaço de cor RGB
- Média do canal de cor H do espaço de cor HSV
- Média do canal de cor S do espaço de cor HSV
- Média do canal de cor V do espaço de cor HSV
- Média do canal de cor Y do espaço de cor YUV
- Média do canal de cor U do espaço de cor YUV
- Média do canal de cor R normalizado (RNorm) do espaço de cor RGB
- Média do canal de cor G normalizado (GNorm) do espaço de cor RGB
- Média do canal de cor B normalizado (BNorm) do espaço de cor RGB
- Entropia do canal de cor R do espaço de cor RGB
- Entropia do canal de cor G do espaço de cor RGB
- Entropia do canal de cor B do espaço de cor RGB
- Entropia do canal de cor H do espaço de cor HSV
- Entropia do canal de cor S do espaço de cor HSV
- Entropia do canal de cor V do espaço de cor HSV
- Entropia do canal de cor Y do espaço de cor YUV
- Entropia do canal de cor U do espaço de cor YUV
- Entropia do canal de cor R normalizado (RNorm) do espaço de cor RGB

- Entropia do canal de cor G normalizado (GNorm) do espaço de cor RGB
- Entropia do canal de cor B normalizado (BNorm) do espaço de cor RGB
- Energia do canal de cor R do espaço de cor RGB
- Energia do canal de cor G do espaço de cor RGB
- Energia do canal de cor B do espaço de cor RGB
- Energia do canal de cor H do espaço de cor HSV
- Energia do canal de cor S do espaço de cor HSV
- Energia do canal de cor V do espaço de cor HSV
- Energia do canal de cor Y do espaço de cor YUV
- Energia do canal de cor U do espaço de cor YUV
- Energia do canal de cor R normalizado (RNorm) do espaço de cor RGB
- Energia do canal de cor G normalizado (GNorm) do espaço de cor RGB
- Energia do canal de cor B normalizado (BNorm) do espaço de cor RGB
- Variância do canal de cor R do espaço de cor RGB
- Variância do canal de cor G do espaço de cor RGB
- Variância do canal de cor B do espaço de cor RGB
- Variância do canal de cor H do espaço de cor HSV
- Variância do canal de cor S do espaço de cor HSV
- Variância do canal de cor V do espaço de cor HSV
- Variância do canal de cor Y do espaço de cor YUV
- Variância do canal de cor U do espaço de cor YUV
- Variância do canal de cor R normalizado (RNorm) do espaço de cor RGB
- Variância do canal de cor G normalizado (GNorm) do espaço de cor RGB
- Variância do canal de cor B normalizado (BNorm) do espaço de cor RGB

APÊNDICE

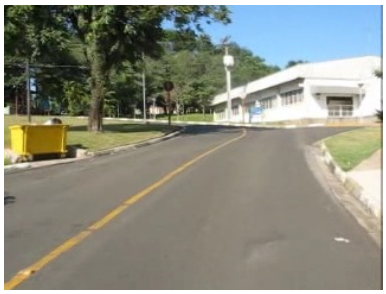
C

Cenários Utilizados

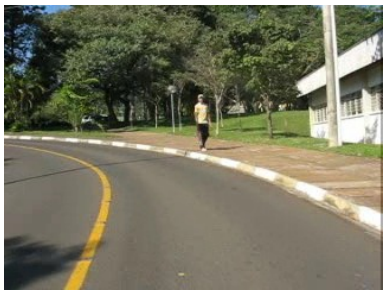
A seguir as imagens utilizadas nos experimentos deste trabalho. As imagens do Cenário 1 foram capturadas de um trajeto realizado no campus 1 da USP de São Carlos. O Cenário 2 é o mesmo trajeto só que em condições de iluminação diferentes - o mesmo trajeto realizado em outro dia. Os Cenários 3 e 4 são um outro trajeto também dentro do campus 1 da USP, onde se diferenciam também nas condições de iluminação. Os Cenários 5 e 6, assim como os outros, pertencem a um outro trajeto no campus 1 variando as condições de iluminação. Portanto temos três cenários em duas condições de iluminação diferentes. As características desses cenários é presença ou não de sombras de árvores sobre as ruas e presença de prédios e carros ladeando a rua, tais características são bem comuns em ambientes urbanos reais. Para finalizar a lista de cenários utilizados, os Cenários 7 e 8 foram obtidos de um percurso no campus 2 da USP de São Carlos. Diferentemente dos demais cenários, estes dois cenários do campus 2 pertencem a mesma condição de iluminação mas trajetos diferentes. Além disso, estes dois últimos cenários são mais simples que os anteriores. O Cenário 7 possui poucas aparições de carros e não tem a ocorrência de sombras e prédios. No Cenário 8 aparecem várias imagens onde a rua está coberta por terra.

(a) Imagem 00

(b) Imagem 01



(c) Imagem 02



(d) Imagem 03

(e) Imagem 04

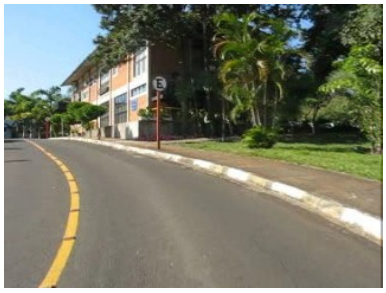
(f) Imagem 05

(g) Imagem 06

(h) Imagem 07

(i) Imagem 08

(j) Imagem 09



(k) Imagem 10

(l) Imagem 11

(m) Imagem 12

(n) Imagem 13

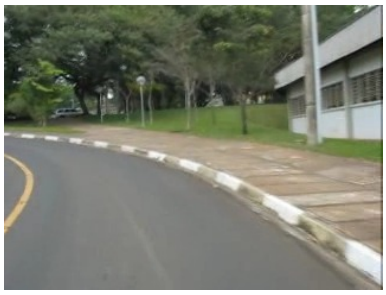
(o) Imagem 14

Figura C.1: Cenário 1

(a) Imagem 00

(b) Imagem 01

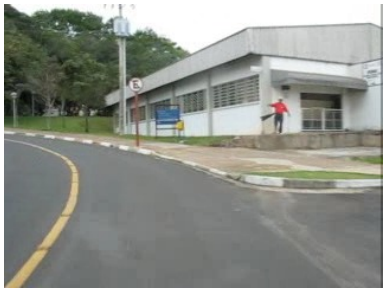
(c) Imagem 02



(d) Imagem 03

(e) Imagem 04

(f) Imagem 05



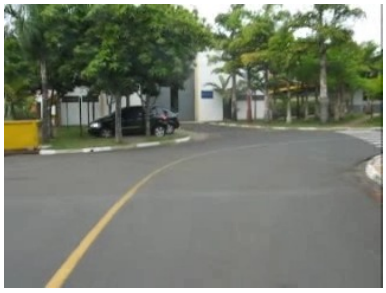
(g) Imagem 06

(h) Imagem 07

(i) Imagem 08

(j) Imagem 09

(k) Imagem 10



(l) Imagem 11

(m) Imagem 12

(n) Imagem 13

(o) Imagem 14

Figura C.2: Cenário 2

(a) Imagem 00

(b) Imagem 01

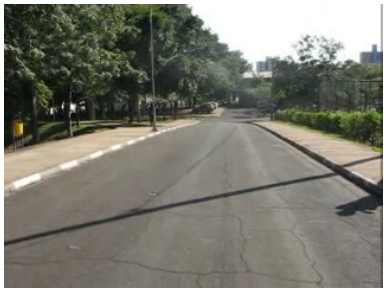
(c) Imagem 02

(d) Imagem 03

(e) Imagem 04

(f) Imagem 05

(g) Imagem 06



(h) Imagem 07

(i) Imagem 08

(j) Imagem 09

(k) Imagem 10

(l) Imagem 11

(m) Imagem 12

(n) Imagem 13

(o) Imagem 14

Figura C.3: Cenário 3

(a) Imagem 00

(b) Imagem 01

(c) Imagem 02

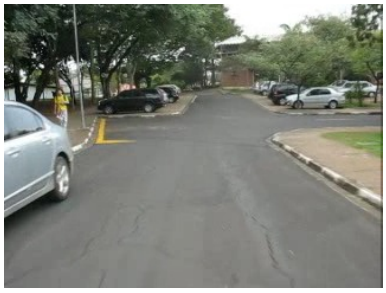
(d) Imagem 03

(e) Imagem 04

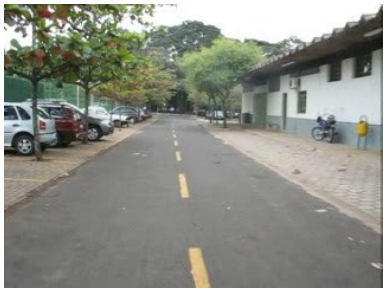
(f) Imagem 05

(g) Imagem 06

(h) Imagem 07



(i) Imagem 08



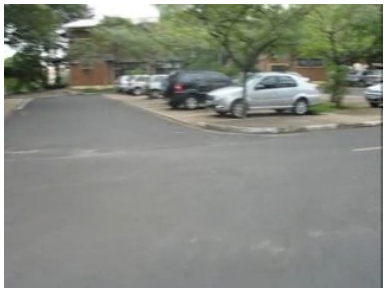
(j) Imagem 09

(k) Imagem 10

(l) Imagem 11

(m) Imagem 12

(n) Imagem 13



(o) Imagem 14

Figura C.4: Cenário 4

(a) Imagem 00

(b) Imagem 01

(c) Imagem 02

(d) Imagem 03

(e) Imagem 04

(f) Imagem 05

(g) Imagem 06

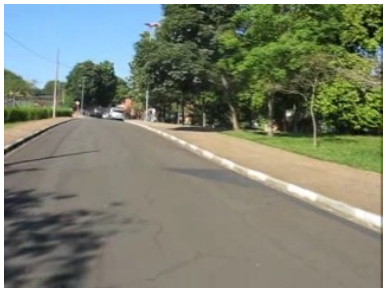
(h) Imagem 07

(i) Imagem 08

(j) Imagem 09

(k) Imagem 10

(l) Imagem 11



(m) Imagem 12

(n) Imagem 13

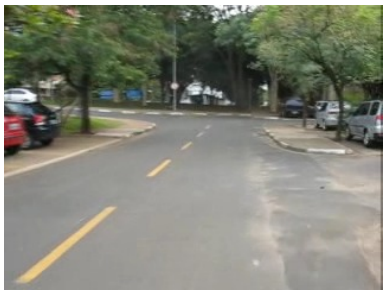
(o) Imagem 14

Figura C.5: Cenário 5

(a) Imagem 00

(b) Imagem 01

(c) Imagem 02



(d) Imagem 03

(e) Imagem 04

(f) Imagem 05

(g) Imagem 06

(h) Imagem 07

(i) Imagem 08

(j) Imagem 09

(k) Imagem 10

(l) Imagem 11

(m) Imagem 12

(n) Imagem 13

(o) Imagem 14

Figura C.6: Cenário 6

(a) Imagem 00

(b) Imagem 01

(c) Imagem 02

(d) Imagem 03

(e) Imagem 04

(f) Imagem 05

(g) Imagem 06

(h) Imagem 07

(i) Imagem 08

(j) Imagem 09

(k) Imagem 10

(l) Imagem 11

(m) Imagem 12

(n) Imagem 13

(o) Imagem 14

Figura C.7: Cenário 7

(a) Imagem 00

(b) Imagem 01

(c) Imagem 02

(d) Imagem 03

(e) Imagem 04

(f) Imagem 05

(g) Imagem 06

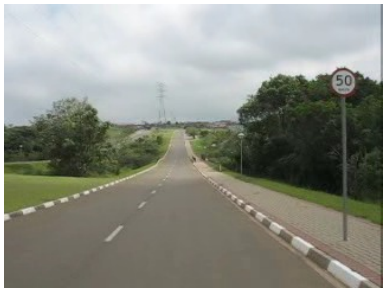
(h) Imagem 07

(i) Imagem 08

(j) Imagem 09

(k) Imagem 10

(l) Imagem 11



(m) Imagem 12

(n) Imagem 13

(o) Imagem 14

Figura C.8: Cenário 8