



**Universidade do Estado do Rio de Janeiro**

**Centro de Ciência e Tecnologia**

**Faculdade de Engenharia**

**Pryscila de Jesus de Sousa**


**Um sistema para o reconhecimento da feição edificação em imagem digital  
com agentes inteligentes**

**Rio de Janeiro**

**2011**

Pryscila de Jesus de Sousa

**Um sistema para o reconhecimento da feição edificação em uma imagem  
digital com agentes inteligentes**



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia de Computação, da Universidade do Estado do Rio de Janeiro. Área de concentração: Geomática.

Orientadora: Prof<sup>ª</sup>. Dr<sup>ª</sup>. Sheila Regina Murgel Veloso

Coorientador: Prof. Dr. Orlando Bernardo Filho

Rio de Janeiro

2011

CATALOGAÇÃO NA FONTE  
UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

S725 Sousa, Priscila de Jesus de.  
Um sistema para o reconhecimento da feição edificação em imagem digital com agentes inteligentes / Priscila de Jesus de Sousa. – 2011. 105f.

Orientadora: Sheila Regina Murgel Veloso.  
Coorientador: Orlando Bernardo Filho.  
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Engenharia de Computação. 2. Agentes inteligentes (Software) - Dissertações. 3. Imagens digitais - Dissertações. I. Veloso, Sheila Regina Murgel. II. Bernardo Filho, Orlando. III. Universidade do Estado do Rio de Janeiro. IV. Título.

CDU 004.8

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta tese, desde que citada a fonte.

---

Assinatura

---

Data

Pryscila de Jesus de Sousa

**Um sistema para o reconhecimento da feição edificação em uma imagem  
digital com agentes inteligentes**

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia de Computação, da Universidade do Estado do Rio de Janeiro. Área de concentração: Geomática.

Aprovado em: 10 de setembro de 2011.

Banca Examinadora:

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Sheila Regina Murgel Veloso (Orientadora)  
Faculdade de Engenharia - UERJ

---

Prof. Dr. Orlando Bernardo Filho (Coorientador)  
Faculdade de Engenharia - UERJ

---

Prof. Dr. João Araújo Ribeiro  
Faculdade de Engenharia - UERJ

---

Prof. Dr. Mauro Antonio Homem Antunes  
Universidade Federal do Rural do Rio de Janeiro – UFRRJ – DENG/IT

Rio de Janeiro

2011

## **DEDICATÓRIA**

À minha família, amigos e a todos os meus professores.

## **AGRADECIMENTOS**

A Deus por me permitir realizar o sonho que foi desenvolver esta dissertação na área que tanto desejava trabalhar.

À minha mãe que sempre me apoiou incondicionalmente, e por todos os seus ensinamentos que me ajudaram a construir um caminho seguro.

À minha orientadora, Prof.<sup>a</sup> Dr.<sup>a</sup> Sheila Regina Murgel Veloso e ao meu co-orientador, Prof. Dr. Orlando Bernardo Filho por todo estímulo, paciência, parceria, confiança e pelos conhecimentos recebidos ao longo destes anos.

Ao Prof. Dr. Mauro Antonio Homem Antunes, que sempre me apoiou e incentivou em minha jornada acadêmica.

À Secretaria Especial de Defesa Civil e Trânsito do Município de Angra dos Reis, pelo apoio e desejo que pudesse me aperfeiçoar e aprimorar conhecimentos.

Aos meus amigos e familiares pelos valores e sabedorias me passados durante toda a minha vida.

## RESUMO

SOUSA, Pryscila de Jesus de. *Um sistema para o reconhecimento da feição edificação em imagem digital com agentes inteligentes*, 2011. 105f. Dissertação (Mestrado em Engenharia da Computação) – Faculdade de Engenharia, Universidade Estadual do Rio de Janeiro, Rio de Janeiro, 2011.

O objetivo desta dissertação foi criar uma nova abordagem para identificar de maneira automática feições do tipo edificação em uma imagem digital. Tal identificação seria de interesse de órgãos públicos que lidam com planejamento urbano para fins de controle da ocupação humana irregular. A abordagem criada utilizou agentes de *software* especialistas para proceder com o processamento da segmentação e reconhecimento de feições na imagem digital. Os agentes foram programados para tratar uma imagem colorida com o padrão *Red, Green e Blue* (RGB). A criação desta nova abordagem teve como motivação o fato das atuais técnicas existentes de segmentação e classificação de imagens dependerem sobremaneira dos seus usuários. Em outras palavras, pretendeu-se com a abordagem em questão permitir que usuários menos técnicos pudessem interagir com um sistema classificador, sem a necessidade de profundos conhecimentos de processamento digital de imagem. Uma ferramenta protótipo foi desenvolvida para testar essa abordagem, que emprega de forma inusitada, agentes inteligentes, com testes feitos em recortes de ortofotos digitais do Município de Angra dos Reis (RJ).

Palavras-Chave: Segmentação de imagem digital; Reconhecimento de feições em imagem digital; Agentes Inteligentes.

## **ABSTRACT**

The purpose of this dissertation has been to create a new approach in order to recognition features of buildings in a digital image in an automatic way. Such recognition features would be interesting of government agencies that deals with urban planning for irregular human occupation control. The approach created has employed specialist software agents to proceed with the segmentation processing and features recognition in the digital images. The agents have been programmed to manipulate colored images with the Red, Green and Blue pattern (RGB). The creation of this new approach has been motivated by the fact of existing segmentation techniques and classification of images greatly depend on its users. In other words, with the approach discussed it has been intended to allow less technical users to interact with a classifier system, without requiring deep knowledge of digital image processing. A prototype tool has been developed to test this approach, which employs in an unusual way, intelligent agents, with tests done in digital orthophotos of the city of Angra dos Reis (RJ).

**Keywords:** Image segmentation; Recognized features; Intelligent Agent.



## LISTA DE FIGURAS

Figura 1. Sistema de coordenadas de uma imagem digital. ....	19
Figura 2. Imagem digital representada por uma matriz $m \times n$ .....	20
Figura 3. Sistema de coordenadas de uma imagem digital. ....	20
Figura 4. Histograma de frequência. ....	21
Figura 5. <i>Pixels</i> adjacentes. (a) Adjacentes por fronteiras e (b) Adjacentes por vértices (JAIN, 1989).....	21
Figura 6. Vizinhanças. (a) vizinhança com 4 <i>pixels</i> e (b) vizinhança com 8 <i>pixels</i> (JAIN, 1989).....	22
Figura 7. (a) Arranjo de <i>pixel</i> ; (b) <i>Pixel</i> central de conectividade; (c) Conectividade mesclada do mesmo <i>pixel</i> (GREEN, 1989).....	23
Figura 8. Gradiente de $f(x, y)$ (GONZALEZ; WINTZ, 2002). ....	25
Figura 9. (a) Imagem ruidosa; (b) Resultado da detecção de contornos (GONZALEZ; WINTZ, 2002). ....	27
Figura 10. Detecção do fundo e do objeto na imagem (MOIGNE; TILTON, 1995).....	28
Figura 11. Geometria do crescimento de regiões (PRATT, 1991).....	29
Figura 12. Algoritmo de Segmentação de Imagens do programa <i>Spring</i> (ESPINDOLA, 2006). ....	30
Figura 13. Relacionamento entre classes informativas e espectrais: I uma informacional para muitas espectrais; II uma espectral para muitas informativas; III mescla de ambas; IV relacionamento biunívoco entre elas (GÓMEZ; SOTOS, 2010).....	33
Figura 14. Classificador de paralelepípedos (CHUVIECO, 2000). ....	37
Figura 15. Extensão a duas bandas do algoritmo de máxima verossimilhança.....	38
Figura 16. Rede neural <i>Back Propagation Neural Network</i> (RICHARDS, 1994). ....	40
Figura 17. Visão geral de agente inteligente (DAVIDSSON, 1992). ....	45
Figura 18. Arquitetura básica de um agente autônomo (DAVIDSSON, 1992). ....	48
Figura 19. Arquitetura básica de um agente reativo simples (RUSSEL; NORVIG, 2003). ....	49
Figura 20. Agente reativo baseado em modelo (RUSSEL; NORVIG, 2003). ....	50
Figura 21. Agente baseado em objetivo (RUSSEL; NORVIG, 2003). ....	50
Figura 22. Agente baseado na utilidade (RUSSEL; NORVIG, 2003). ....	51
Figura 23. Agente com aprendizagem (RUSSEL; NORVIG, 2003).....	51
Figura 24. Visão de um Sistema Multiagente (HUBNER; SICHMAN, 2003).....	54
Figura 25. Visão das interações de um agente sobre o ambiente .....	56
Figura 26. Fluxo de trabalho de agentes inteligentes aplicados ao reconhecimento de feições (ORTEGA, 2008). ....	58

Figura 27. Arquitetura do Sistema de Classificação de Imagem.....	62
Figura 28. (a) Recorte 1, para área denominada A e (b) Recorte 2, para área denominada B	63
Figura 29. Evolução da ocupação da área central (Centro, ao sul e Japuíba, ao norte) do município de Angra dos Reis: (a) aerofotograma de 1962 (LASA), apresentando ocupação restrita ao Centro; (b) aerofotograma de 1991 (AEROIMAGEM); e (c) Imagem de satélite, IKONOS III de 2007, apresentando uma ocupação urbana em Planície Costeira. (Acervo da Secretaria Municipal de Planejamento de Angra dos Reis). ....	64
Figura 30. Ocupação humana em área de encostas acerca do centro comercial de Angra dos Reis. ....	65
Figura 31. Representação da inferência do agente especialista em textura. ....	66
Figura 32. Diagrama de atividades dos agentes especialistas em textura. ....	67
Figura 33. Diagrama de atividades dos agentes especialistas em geometria para proporção do objeto edificação.....	68
Figura 34. Diagrama de atividades dos agentes especialistas em geometria para área padrão do objeto edificação.....	69
Figura 35. Agente Especialista em Segmentação: (a) Imagem digital onde os <i>pixels</i> são representados pelos valores de cores RGB e (b) Imagem digital segmentada, onde as áreas brancas representam os <i>pixels</i> próximos homogêneos e as linhas pretas as bordas destas áreas . ....	70
Figura 36. Diagrama de atividades dos agentes especialistas em segmentação.....	70
Figura 37. Diagrama de atividades dos agentes especialistas em conflito. ....	72
Figura 38. A interface do sistema multiagente no ambiente Netlogo. ....	73
Figura 39. Representação dos sliders para configuração da faixa de valores de RGB.....	74
Figura 40. Representação do mundo (imagem digital).....	74
Figura 41. Botões para execução dos agentes especialistas em textura. ....	75
Figura 42. Representação dos sliders que configuram os parâmetros área padrão e geometria utilizados pelo agente geometria para a classificação das edificações.....	75
Figura 43. Representação do slide de limiar e do botão para execução do agente especialista em segmentação.....	75
Figura 44. Representação dos botões carregar imagem, e conflito que executa o agente especialista em conflito. ....	76
Figura 45. Representação do botão salvar imagem. ....	76
Figura 46. Resultado do agente para segmentação com limiares iguais a 16, 25, 27 e 35 respectivamente. ....	79
Figura 47. Resultado para o reconhecimento da feição vias, vegetação e edificação da imagem do recorte 1. ....	80
Figura 48. Resultado do agente especialista segmentação: (a) Limiar = 20, (b) Limiar = 30 e (c) Limiar = 45.....	81
Figura 49. Resultado para o reconhecimento da feição vias, vegetação e edificação da imagem do recorte 2. ....	82

Figura 50. Resultado da classificação final. (a) imagem do recorte 1 e (b) imagem do recorte 2. ....	83
Figura 51. Resultados da classificação de imagem por máxima verossimilhança realizado pelo programa ArcGis 10: (a) Recorte 1 e (b) Recorte 2. ....	84
Figura 52. Resultados da segmentação e classificação de imagem realizada pelo programa <i>Spring</i> 5.1.8: (a) Segmentação com limiar de área = 10 e limiar de similaridade = 25 (b) Classificação por regiões através do algoritmo Bhattacharya e (c) Classificação por <i>pixel</i> através do algoritmo máxima verossimilhança. ....	85
Figura 53. Resultados da segmentação e classificação de imagem realizada pelo programa <i>Spring</i> 5.1.8: (a) Segmentação com limiar de área = 10 e limiar de similaridade = 25 (b) Classificação por regiões através do algoritmo Bhattacharya e (c) Classificação por <i>pixel</i> através do algoritmo máxima verossimilhança. ....	86

## LISTA DE TABELAS

Tabela 1 - Exemplo de matriz de confusão .....	42
Tabela 2 - Atitudes de informação e pró-atitudes de um agente .....	52
Tabela 3 - Análise numéricas dos resultados no protótipo. ....	82
Tabela 4 - Análise numéricas dos resultados no ArcGis. ....	84
Tabela 5 - Análise numéricas dos resultados no <i>Spring</i> . ....	86
Tabela 6 - Resultados dos testes nos programas NetLogo (Protótipo), ArcGis e Spring. ....	87

# SUMÁRIO

	<b>INTRODUÇÃO</b> .....	12
1	<b>OBJETIVO</b> .....	13
1.1	<b>Motivação</b> .....	13
1.2	<b>Justificativa</b> .....	15
1.3	<b>Trabalhos Relacionados</b> .....	16
1.4	<b>Organização do Documento</b> .....	17
2	<b>FUNDAMENTAÇÃO TEÓRICA</b> .....	18
2.1	<b>Representação do Espaço RGB</b> .....	18
2.1.1	<u>Relação de Vizinhança entre <i>Pixel</i></u> .....	21
2.1.2	<u>Conectividade entre <i>Pixels</i></u> .....	22
2.2	<b>Segmentação de Imagem</b> .....	23
2.2.1	<u>Deteção de Bordas</u> .....	25
2.2.2	<u>Limiarização ( Thresholding)</u> .....	27
2.2.3	<u>Crescimento de Regiões</u> .....	29
2.3	<b>Classificação de Imagem Digital</b> .....	31
2.3.1	<u>Fase de Treinamento</u> .....	32
2.3.2	<u>Método Supervisionado</u> .....	34
2.3.3	<u>Método Não Supervisionado</u> .....	35
2.3.4	<u>Fase de Atribuição</u> .....	35
2.3.4.1	Classificador de Distância Mínima .....	36
2.3.4.2	Classificador de Paralelepípedo .....	37
2.3.4.3	Classificador de Máxima Verossimilhança .....	38
2.3.4.4	Classificador de Mahalanobis .....	39
2.3.4.5	Outros Critérios de Classificação .....	40
2.3.4.6	Fase de Verificação .....	42
2.4	<u>Abordagem Proposta</u> .....	44
2.5	<b>Agentes Inteligentes</b> .....	45
2.5.1	<u>Características dos Agentes Inteligentes</u> .....	47
2.5.2	<u>Tipos de Agentes Inteligentes</u> .....	48
2.5.3	<u>Arquitetura dos Agentes Inteligentes</u> .....	49
2.5.4	<u>Estados Mentais</u> .....	52

2.5.5	<u>Sistema Multiagentes</u> .....	54
2.5.6	<u>Aspectos Fundamentais</u> .....	55
2.5.7	<u>Linguagem dos Agentes</u> .....	56
2.5.8	<u>Tipos de Ambientes</u> .....	56
2.5.9	<u>Agentes Inteligentes Aplicados no Reconhecimento de Feições</u> .....	58
2.6	<b>NetLogo</b> .....	59
3	<b>UM PROTÓTIPO PARA O RECONHECIMENTO DA FEIÇÃO EDIFICAÇÃO</b> .....	61
3.1	<b>Componentes do Sistema de Classificação de Imagem</b> .....	61
3.2	<b>Contextualização dos Cenários</b> .....	63
3.3	<b>Aspectos do Protótipo do Sistema</b> .....	65
3.4	<b>Interface do Protótipo do Sistema</b> .....	73
4	<b>RESULTADOS E ANÁLISES</b> .....	77
4.1	<b>Imagens para Testes</b> .....	77
4.2	<b>Resultados</b> .....	79
4.3	<b>Avaliação do Protótipo</b> .....	83
5	<b>CONCLUSÕES</b> .....	88
5.1	<b>Trabalhos Futuros</b> .....	90
	<b>REFERÊNCIAS</b> .....	91
	<b>APÊNDICE A – Pseudocódigo</b> .....	95
	<b>APÊNDICE B – Código Fonte</b> .....	96

## INTRODUÇÃO

O reconhecimento de feições em uma imagem digital é um dos passos mais importantes do processamento de imagens, já que as bordas e contornos representam em uma imagem os limites dos objetos nela contidos. O processo de reconhecimento de feições é baseado na separação de regiões similares de uma imagem em termos de intensidades e padrões de tons de cinza do *pixel*. Além disso, esta técnica apresenta informações essenciais de um objeto na imagem de interesse imediato em estudos urbanos. Uma característica altamente desejável para a extração de feições é que a mesma represente somente as bordas físicas pertinentes à sua classe, evitando as espúrias provenientes de ruído e textura da imagem. Um detector ideal de feições que permita apenas a detecção de suas bordas físicas, sem fragmentação e deslocamento da sequência de pontos que representa o contorno do objeto, não existe, tornando-se necessário fazer uso de estratégias que permitam obter resultados mais próximos possíveis do ideal desejado.

O presente trabalho apresenta a possibilidade do reconhecimento das feições das edificações, em ortofoto digital através de um sistema multiagente. Um sistema multiagente é composto por agentes inteligentes que atuam em um ambiente e são capazes de tomar ações autônomas de acordo com o ambiente e suas metas. Estes componentes autônomos interagem entre si possibilitando que seus objetivos particulares se interrelacionem. O sistema desenvolvido fornece uma interface de fácil interação para usuários comuns, sem a obrigatoriedade do conhecimento prévio dos parâmetros técnicos para a execução dos algoritmos envolvidos nas diferentes análises que o sistema oferece. Suas funções permitem analisar a ocupação e dinâmica do espaço urbano, além de oferecer suporte para a gestão pública no que se refere ao controle do crescimento urbano. Neste projeto, dois recortes distintos das ortofotos digitais do município de Angra dos Reis foram utilizados como ambientes de teste, onde os agentes se encontram que descrevem o espaço geográfico urbano. Na implementação do sistema multiagente, utilizou-se a plataforma NetLogo 4.1.3, um programa particularmente direcionado para o desenvolvimento de modelos e simulações de fenômenos naturais, sociais e biológicos baseados em agentes inteligentes.

## 1 OBJETIVO

O principal objetivo deste trabalho é implementar um protótipo com arquitetura baseada em sistema multiagentes, que possibilitará a segmentação e posterior reconhecimento das feições das edificações de uma imagem digital. No entanto, para alcançar esse objetivo será necessário o cumprimento dos seguintes objetivos específicos:

- Analisar os conceitos e regras que envolvem as técnicas de segmentação e classificação de imagens aéreas e o conhecimento de agentes inteligentes;
- Desenvolver sete agentes inteligentes distintos, os quais farão parte do sistema multiagente que percorrerá o ambiente da imagem aérea. Nesta etapa, foi definido o comportamento dos agentes e a forma como eles atuam no ambiente;
- Montar a estratégia do sistema multiagente, sabendo-se que o objetivo do sistema é a tomada de decisão para a correta classificação dos segmentos detectados na imagem aérea;
- Definir e implementar a arquitetura interna dos agentes que terão ao mesmo tempo, as camadas de reconhecimento e interpretação/validação de padrões;
- Definir e implementar procedimentos para controlar a interação entre os agentes;
- Realizar a simulação do reconhecimento das feições. A simulação será baseada no paradigma da programação multiagente com o uso da plataforma Netlogo.
- Analisar a colaboração entre os agentes. A análise será essencial para verificar o reconhecimento das feições por parte dos agentes inteligentes no ambiente da imagem, isto é, se eles atingiram ou não suas metas individuais e globais, ou se existiram evoluções dentro do ambiente.

### 1.1 Motivação

O planejamento e a administração urbana possuem suas decisões baseadas em um processo de grande dependência de dados espaciais, sendo que uma excelente fonte para essa atividade é o Cadastro Técnico Imobiliário. Esse possui caráter fundamental no planejamento e na gestão territorial, pois oferece as ferramentas e a precisão necessárias para elaboração de planos urbanísticos e monitoramento da ocupação em assentamentos humanos. As mudanças das tradicionais tendências da dinâmica da concentração e distribuição da população urbana



no país são observadas por especialistas e exigem soluções em políticas públicas em consonância com a gestão municipal.

O atual cenário do município de Angra dos Reis retrata bem essa nova realidade no processo de expansão urbana. Existe um cadastro imobiliário cuja utilização se restringe apenas a uma base de dados para cálculo de tributos, que não fornece subsídio necessário para a caracterização dos imóveis, com a finalidade da implementação de uma base cartográfica consistente.

Como solução para esta problemática pertinente ao município, é necessário resgatar outras escalas de apreensão do processo de urbanização. Esse trabalho busca valorizar a importância de investimentos em tecnologias digitais de geoprocessamento, tanto em processos de aquisição de dados espaciais, como também na geração de informações precisas e documentos cartográficos adequados para a gestão territorial. Com o avanço tecnológico durante os últimos 50 anos, as imagens de satélite e as fotografias aéreas têm sido cada vez mais comumente utilizadas em diversas aplicações relacionadas a problemas urbanos. Alguns municípios e instituições governamentais têm construído bases de dados compostas por imagens digitais a fim de servirem de fontes seguras para manterem atualizadas as suas bases cartográficas vetoriais. Em caso de detecção de mudanças que ocorrem em cenários associados a determinados recortes do espaço geográfico, as imagens digitais podem atualizar de forma eficiente e com boa frequência a base cartográfica do município.

O presente trabalho tem como motivação a utilização de imagens digitais como solução para a atualização da base de dados cadastrais. Nesta atualização, a identificação dos imóveis e a evolução do perímetro urbano são feitas de forma automática. Essas informações oferecem suporte para o controle do processo de crescimento urbano desordenado e fomenta a base cadastral de imóveis do município. Diante do aumento da quantidade de imagens digitais disponíveis na base de dados espaciais, que são utilizadas como fonte de análise do planejamento e da administração urbana, as informações fornecidas ao cadastro técnico imobiliário podem ser mais facilmente obtidas com uso automático de tal análise sobre essas imagens digitais.

A extração de informações das imagens através do processamento digital é hoje um vasto campo de estudo e pesquisa em várias disciplinas, com múltiplas aplicações. Neste sentido, fotogrametristas, matemáticos, físicos, cientistas da computação e outros continuam a investigar questões para automatizar o reconhecimento semântico de objetos em imagens. No entanto, grandes realizações já foram alcançadas. A detecção automática de feições de imagens tem uma grande tradição no campo da computação visual e existem muitos métodos

para esta finalidade. Infelizmente, apesar da gama de métodos, não há um método universal para a detecção automática de feições.

Uma imagem proporciona pouca informação para uma interpretação visual das feições que compõem o cenário. Um sistema de classificação de imagem deve, em primeira etapa, extrair de forma eficiente e robusta determinadas características que proporcionem informações da realidade do cenário. A proposta do Sistema desenvolvido neste trabalho é cumprir os requisitos mencionados anteriormente e apresentar um sistema de classificação adaptável a qualquer tipo de cenário de imagem.

## 1.2 Justificativa

Uma nova técnica para a segmentação e reconhecimento de feições em imagem digital é proposta neste trabalho. É apresentado um protótipo utilizando agentes inteligentes, baseado na técnica de crescimento de regiões, de modo a aprimorar o obtido no reconhecimento de feições de uma imagem. Dentro dos diferentes tipos de algoritmos de segmentação, o crescimento por regiões é comumente utilizado. Suponha que um *pixel* semente tenha *pixels* adjacentes com propriedades semelhantes. O algoritmo parte da semente e aplicando certas condições de inclusão, crescem e admitem novos *pixels* na região, e assim delimitam uma região homogênea. Os problemas que surgem são a escolha da semente e a condição de parada para obter a melhor região. Em geral, os algoritmos de crescimento de regiões fixam essas condições, fazendo com que um algoritmo seja útil para determinados tipos de imagens, mas não para aplicação em geral.

Este trabalho apresenta um protótipo de um sistema multiagente segmentador e identificador de feições em imagem digital, baseado nos conceitos da técnica de segmentação e classificação por meio de crescimento de regiões, onde a condição de inclusão de *pixel* à região depende dos parâmetros cor, textura e geometria analisados por agentes especializados sobre a cena da imagem digital. A grande vantagem deste sistema é a de ser independente do tipo de imagem utilizada, diferentemente do que ocorre em outros métodos de segmentação e classificação de imagem, embora por questões de desempenho as imagens empregadas nos testes continham um mesmo tamanho em torno de 300 x 300 *pixels* aproximadamente.

### 1.3 Trabalhos relacionados

O avanço no desenvolvimento computacional permite encontrar novas aplicações para o processamento digital de imagens. A segmentação e a classificação de imagens digitais é uma parte importante para muitas tarefas de análises e processamento digital.

Atualmente, tem-se gerado uma grande quantidade de trabalhos que apresentam técnicas, modelos e algoritmos para a segmentação e reconhecimento de feições de uma imagem, como se destaca a seguir alguns trabalhos relacionados.

Khouadjia, Khanfouf e Meshoul (2001) utilizaram uma arquitetura multiagente para segmentação de imagem digital. Cada agente apresenta vários comportamentos reativos em resposta ao estímulo do ambiente. Os agentes podem migrar-se, reproduzirem-se ou propagarem-se no interior da imagem. O projeto foi implementado na plataforma multiagente NetLogo, devido aos benefícios oferecidos por este programa. Os resultados obtidos foram muito satisfatórios e demonstraram a eficácia da abordagem.

Angelotti (2001) implementou dois agentes inteligentes no processo de reconhecimento de padrões aplicado ao projeto Multiteck 1, que define uma arquitetura de agentes autônomos para o tratamento automático de cheques bancários brasileiros manuscritos. Esses dois agentes foram implementados em duas camadas. A primeira corresponde aos algoritmos de reconhecimento de padrões aplicados diretamente sobre os segmentos de imagens. A segunda corresponde aos mecanismos de raciocínio aplicados sobre as informações provenientes da primeira camada para validá-las ou interpretá-las. Esta interpretação envolve também informações provenientes de outros agentes. O que pode gerar inconsistências. Este problema foi tratado através de conceitos e operadores da lógica evidencial paraconsistente.

Lima (2005) utilizou agentes inteligentes para separação de dígitos manuscritos. A imagem é pré-processada através da aplicação de um algoritmo de esqueletização. Em seguida, dois agentes começam a analisar a imagem. O primeiro varre seus *pixels* de cima para baixo, buscando por um ponto de corte, enquanto que o segundo faz o mesmo, porém, analisando a imagem de baixo para cima. Após definir seus pontos de corte, os agentes entram em processo de negociação e resolução de conflitos, visando obter um ponto de consenso para o corte da imagem. Caso cada um tenha selecionado um ponto distinto, então, o primeiro agente segmenta a imagem e aplica um algoritmo de restauração para o reposicionamento de segmentos curtos desconexos que eventualmente a imagem segmentada possa conter.

Walsh. et al. (2010) desenvolveram um modelo baseado em agentes com uma abordagem analítica que oferece a capacidade de informar através dos experimentos, questões fundiárias importantes para a ciência e a sociedade. Utilizando uma paisagem, onde um selecionado conjunto de elementos-chave sociais, geográficos e ecológicos é espacialmente organizado, foi descrita a dinâmica do uso e cobertura da Terra que pode ser examinada através de um modelo baseado em agente que serve como ferramenta educacional útil em sala de aula, sala de reuniões e fóruns públicos.

A necessidade de criar novas técnicas de extração de feições de imagens que sejam eficientes na sua aplicação em qualquer tipo de imagem digital tem gerado novos campos de desenvolvimento no processo de reconhecimento de feições, e uma dessas novas implementações é a que será discorrido neste trabalho: a utilização de agentes inteligentes como forma de utilizar-se de parâmetros abrangentes que incluem a base do conhecimento dos agentes que pode conter dados socioeconômicos, físicos, estatísticos etc. e um ambiente, que é a imagem digital onde os agentes interagem.

#### **1.4 Organização do Documento**

Esta dissertação está organizada em quatro capítulos. O primeiro capítulo proporciona uma visão geral dos conceitos necessários para o entendimento do escopo do trabalho desenvolvido, apresentando os princípios e fundamentos de segmentação e classificação de imagens, incluindo a definição de agentes Inteligentes que compõem um sistema multiagente, bem como suas características e funcionalidades. No segundo capítulo, apresenta-se alguns aspectos da implementação e da metodologia utilizada no desenvolvimento do protótipo do sistema multiagente especialista em reconhecimento de feições em imagens digitais. No terceiro capítulo, estão os resultados obtidos, bem como alguns testes e análises comparativas para validação do protótipo. Finalmente, o quarto capítulo, apresenta as conclusões para esta dissertação e recomendações para trabalhos futuros.

## 2 FUNDAMENTOS TEÓRICOS

Este capítulo proporciona uma visão geral dos conceitos necessários para o entendimento do escopo deste trabalho. Apresenta-se, em primeiro lugar, o espaço de representação das bandas RGB, e as características de uma imagem digital. Em seguida, faz-se uma abordagem sobre as principais técnicas de segmentação de imagem, assim como evidencia-se a importância da segmentação para a posterior classificação dos objetos reconhecidos em uma imagem. Em seguida, são descritos os principais algoritmos para a classificação de imagens e apresenta-se considerações que indicam os principais problemas no processo de classificação. Por fim, aponta-se os principais conceitos dos sistemas multiagentes, como objetivo de justificar a implementação de uma nova técnica para a construção de um sistemas de reconhecimento de feições em imagens digitais.

### 2.1 Representação do Espaço RGB

A análise digital de feições obteve nos últimos anos um grande desenvolvimento das técnicas direcionadas a análise de dados multidimensionais adquiridos por diversos tipos de sensores. O conjunto destas técnicas é denominado de processamento digital de imagens.

O processamento digital de imagens é o conjunto de procedimentos relativos à manipulação e análise de imagens por meio do computador (JENSEN, 1986). De acordo com Gonzalez e Wintz (1977), o objetivo do processamento digital de imagem é ser um conjunto de procedimentos, que melhoram a informação representada em forma de *pixels* para a interpretação do olho humano, bem como para o reconhecimento das feições de uma imagem digital.

Antes de abordar as técnicas de segmentação e classificação de imagens digitais, convém destacar que a representação da cor não é um problema trivial e que esta influi notavelmente no modo e na eficiência da implementação destas técnicas.

O espaço RGB, que representa a banda R (cor vermelha), G (cor verde) e B (cor azul), é o espaço de cores mais utilizado entre os modelos de cores existentes, especialmente para os dados de 8 bits. A representação do espaço RGB é feita por meio de um cubo (Figura 1) onde uma cor define uma mescla de valores de intensidade de três cores primárias, vermelho, verde

e azul. Uma cor é descrita por uma tupla de três coordenadas no cubo. A cor preta é representada por RGB (0, 0, 0) e a cor branca por RGB (255, 255, 255). O valor da faixa acromática em tons de cinza está representado pela diagonal do cubo (JAIN, 1989).

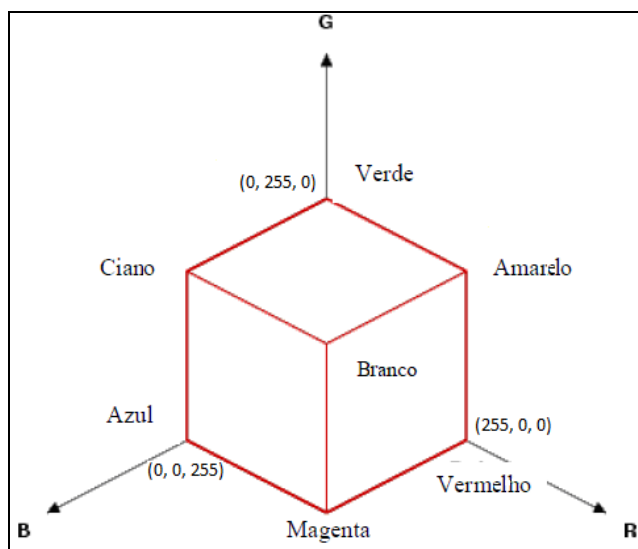


Figura 1. Sistema de coordenadas de uma imagem digital.

Para representar uma imagem em sensoriamento remoto é necessário um grande número de dados. Estes dados podem ser manipulados no formato digital, objetivando a extração de informações pertinentes às imagens. Os pontos imageados pelos sensores correspondem a uma área mínima denominada *pixel*, que está geograficamente identificado, e para o qual são registrados valores digitais referentes a intensidade de energia refletida em faixas (bandas) bem definidas do espectro eletromagnético. O espectro eletromagnético é o intervalo completo da radiação eletromagnética, que contém desde as ondas de rádio, as microondas, o infravermelho, a luz visível, os raios ultravioleta, os raios X, até à radiação gama.

O nível de brilho, ou o valor de cada *pixel* é quantizado em código binário inteiro positivo (a luminosidade não pode ser negativa). O Número de níveis de quantização ( $L$ ) é determinado pela relação:

$$L = 2^B \quad (1)$$

onde  $B$  representa o número de bits do *pixel*.

Isto é, uma imagem com 8 bits de quantização em intensidade pode-se representar 256 níveis de cinza.

Uma imagem é representada por uma função bidimensional, da intensidade de luz refletida ou emitida por uma cena, na forma  $I(x, y)$ , onde  $x$  e  $y$  são coordenadas espaciais.

(SABINS, 1987). Uma matriz de  $x$  linhas e  $y$  colunas definida pelas coordenadas espaciais do *pixel* matriz pode representar uma imagem. Assim, uma imagem digital é uma função  $I(x, y)$ , definida por uma grade regular de  $m$  linhas e  $n$  colunas, representada por  $I(i, j)$ , onde:  $0 \leq i \leq m - 1$  (Figura 2).

$0 \leq j \leq n - 1$  para  $I$  variando de  $[0, k-1]$ ;

Onde,  $k$  = níveis de cinza.

Assim, uma imagem digital pode ser representada por uma matriz  $I$  de tamanho  $m \times n$  dada por:

$$I = \begin{matrix} I(1,1) & I(1,2) & \dots & I(1,n) \\ I(2,1) & I(2,2) & \dots & I(2,n) \\ \vdots & \vdots & & \vdots \\ I(m,1) & I(m,2) & \dots & I(m,n) \end{matrix}$$

Figura 2. Imagem digital representada por uma matriz  $m \times n$ .

A Figura 3 demonstra o sistema de coordenadas comumente utilizado para representar uma imagem digital.

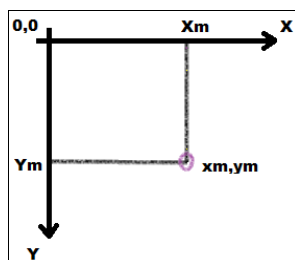


Figura 3. Sistema de coordenadas de uma imagem digital.

A representação digital das imagens multiespectrais é mais complexa, pois para cada coordenada  $(x, y)$  haverá um conjunto de valores de nível de cinza. O *pixel* é então representado por um vetor, com tantas dimensões quantas forem as bandas espectrais. A banda espectral é definida pelo intervalo entre dois comprimentos de onda no espectro eletromagnético.

No caso da imagem RGB, cada banda R (cor vermelha), G (cor verde) e B (cor azul) da imagem possui uma variação de tons de cinza de 0 a 255. O histograma de frequência é a forma mais utilizada para representar a distribuição dos níveis de cinza de uma imagem (Figura 4). O histograma fornece informações sobre quantos *pixels* na imagem possuem determinado valor de cinza (para uma imagem de 8 bits varia de 0 a 255). Para o aumento do contraste e realce de uma imagem são importantes os dados discretos representados nos histogramas (RICHARDS; JIA, 1994).

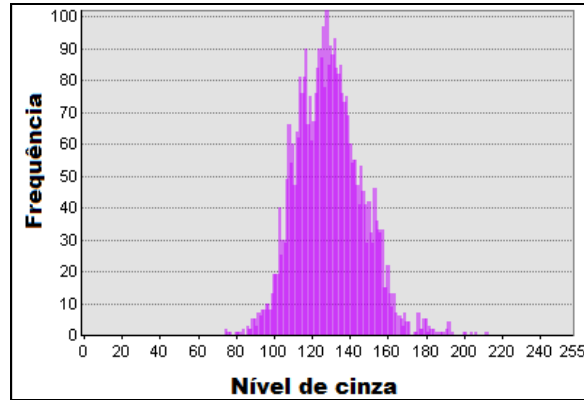


Figura 4. Histograma de frequência.

### 2.1.1 Relação de Vizinhança entre Pixels

De acordo com Gonzalez e Woods (2002), um *pixel*  $p$  na coordenada  $(x, y)$  tem 4 vizinhos horizontais e 4 vizinhos verticais, cujas coordenadas são dadas por:

$$(x+1, y), (x-1, y), (x, y+1), (x, y-1) \quad (2)$$

O mesmo ocorre com a vizinhança diagonal com o ponto, cujas coordenadas são:

$$(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1) \quad (3)$$

Para definir de forma adequada o conceito de vizinhança, é necessário revisar o de adjacência. Dois *pixels* são adjacentes se, e somente se, têm em comum uma de suas fronteiras, ou pelo menos um de seus vértices em comum. A Figura 5 mostra os *pixels* adjacentes.

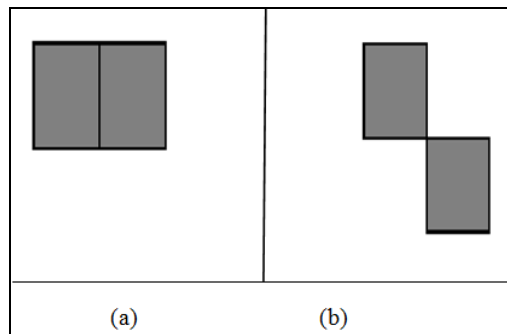


Figura 5. *Pixels* adjacentes. (a) Adjacentes por fronteiras e (b) Adjacentes por vértices (JAIN, 1989).

Dois *pixels* são vizinhos se cumprem a definição de adjacência. Se os *pixels* compartilham uma de suas fronteiras, se diz que os mesmos são vizinhos diretos; se somente se tocam em um de seus vértices, se chamam vizinhos indiretos.

Uma vizinhança de um *pixel*  $p_0$  denominada como  $V_p$ , é uma submatriz MKL de tamanho  $K \times L$ , com  $K$  e  $L$  inteiros ímpares, contida na matriz imagem  $(i_{MN})$ , a qual é formada por um número finito de *pixels* vizinhos ou não de  $p_0$ .



$$V_p = \{p: p \in M_{KL}\}; M_{KL} \subset i_{MN}; K=L=3, 5, 9 \quad (4)$$

Para ilustrar as definições anteriores, pode-se observar, na Figura 6, vizinhança de 4 ( $V_4$ ) e vizinhança de 8 ( $V_8$ ), a primeira formada, por *pixels* que são vizinhos diretos, enquanto que a vizinhança de 8 está formada tanto por vizinhos diretos como por indiretos.

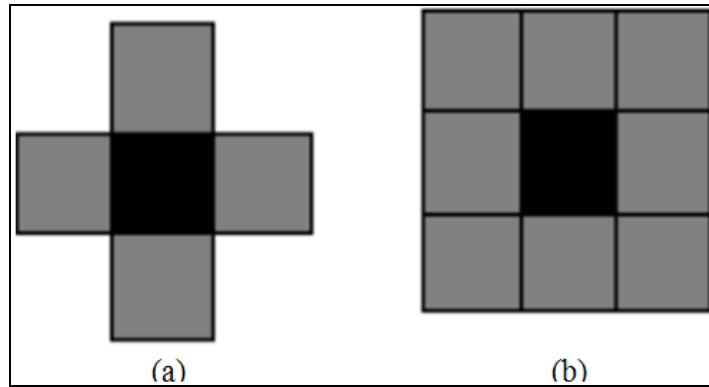


Figura 6. Vizinhanças. (a) vizinhança com 4 *pixels* e (b) vizinhança com 8 *pixels* (JAIN, 1989).

No desenvolvimento das técnicas de processamento de imagens que envolvem a análise de uma determinada região da cena, é possível encontrar vizinhança com 5x5 até com 9 x 9 *pixels*; basicamente a definição das dimensões da matriz vizinhança depende da técnica que se está desenvolvendo.

### 2.1.2 Conectividade entre *Pixels*

A conectividade entre os *pixels* é um conceito utilizado para estabelecer os limites dos objetos e regiões que compõem uma imagem. Para estabelecer a conectividade entre os *pixels*, é necessário determinar se são adjacentes em sentido específico (se tem 4 vizinhos) e se seus níveis de cinza satisfazem um critério especificado de similaridade (se são iguais). Por exemplo, em uma imagem binária, com valores 0 e 1, dois *pixels* podem ter 4 vizinhos, mas somente se consideram conectados se têm os mesmos valores.

Considerando  $V$  como valores de níveis de cinza utilizados para definir conectividade e  $N_D(p)$  como a matriz da vizinhança; pode-se exemplificar, através de uma imagem binária,  $V = \{1\}$ , para a conectividade de *pixel* com valor de faixa de intensidade entre 32 a 64, seria  $V = \{32, 33, 34, 35, \dots, 63, 64\}$ . Existem três tipos de conectividade:

- Conectividade de 4 *pixels*: Dois *pixels*,  $p$  e  $q$ , com valores de  $V$  estão conectados se  $q$  pertence a  $N_4(p)$ .

- Conectividade de 8 *pixels*: Dois *pixels*,  $p$  e  $q$ , com valores de  $V$ , estão conectados se  $q$  pertence a  $N_8(p)$ .
- Conectividade mesclada: Dois *pixels*,  $p$  e  $q$ , com valores de  $V$ , estão conectados mesclados se:
  - $q$  pertence a  $N_4(p)$ , ou
  - $q$  pertence à  $N_D(p)$  e  $N_4(p) \cap N_4(q)$  está vazio.

A conectividade mesclada é uma modificação da conectividade de 8 *pixels*, e se introduz para eliminar as conexões multi-trajetórias (Figura 7). Um *pixel*  $p$  é adjacente a  $q$  se estão conectados. Uma trajetória de *pixel*  $p$  com coordenadas  $(x, y)$ , a um *pixel*  $q$  com coordenadas  $(s, t)$  é uma sequência de distintos *pixels* com coordenadas:

$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n) \quad (5)$$

$$\text{Onde } (x_0, y_0) = (x, y) \text{ e } (x_n, y_n) = (s, t) \quad (6)$$

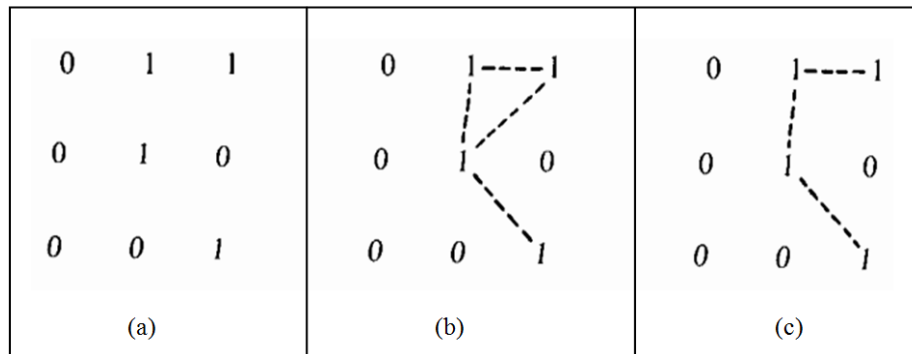


Figura 7. (a) Arranjo de *pixel*; (b) *Pixel* central de conectividade; (c) Conectividade mesclada do mesmo *pixel* (GREEN, 1989)

## 2.1 Segmentação de Imagem

A forma de um objeto pode ser descrita por suas bordas ou pela região que ocupa. A representação de um objeto baseada em bordas tem como requisito a detecção de seus contornos; para a representação baseada em regiões, se realiza a segmentação de várias regiões homogêneas depois de executada a detecção dos contornos dos objetos. Essas regiões têm por finalidade modificar a representação dos objetos na imagem e facilitar suas análises. Após a segmentação, a imagem é composta por um conjunto de contornos. A cor, textura e

intensidade de cinza de um *pixel* são características em comum do contorno de uma região (MELENDEZ; ALTAMIRANO, 2003).

A correta segmentação da imagem determina o sucesso ou falha nos processos de análise e classificação da imagem. Não existe um modelo formal para a segmentação, sendo esse processo essencialmente empírico e, em geral, ajustável aos diferentes tipos de imagens, com limiares definidos conforme a complexidade dos alvos investigados (ANTUNES, 2003).

O uso da segmentação anterior à classificação de imagens supera as limitações apresentadas com o processo de análise pontual (*pixels* analisados de forma isolada), que se baseia unicamente em atributos espectrais. Os elementos analisados e utilizados na classificação são as regiões resultantes da aplicação do segmentador utilizado na definição do espaço de atributos da classificação (VENTURIERI; SANTOS, 1998). De uma maneira geral, a eficiência dos algoritmos de segmentação de imagens está vinculada à extração automática dos objetos de interesse. Neste sentido, há métodos que são adaptados a tipos particulares de aplicação, sendo que não existem algoritmos de segmentação convenientes a todos os tipos de imagens (BEAUCHEMIM; THOMSOM, 1997). Apesar da crescente variedade de métodos destinados a essa aplicação, o método de segmentação por crescimento de regiões é mais comumente utilizado, visto que considera a característica espacial dos dados e garante a formação de segmentos fechados (TILTON; LAWRENCE, 2000). Alguns produtos de *software* de segmentação de imagem de aplicação em sensoriamento remoto implementam o algoritmo de crescimento de regiões em suas rotinas, podendo-se citar os programas *Spring* e o *e-Cognition®* (BAATZ; SCHAPE, 2000). Existem vários métodos para executar a segmentação de uma imagem, como a divisão e agrupamento de regiões (*Split and Merge*), o crescimento de regiões (*Region Growing*), entre outros.

Os algoritmos de segmentação permitem achar diferenças entre dois ou mais objetos e distinguir as partículas umas das outras e do fundo. Esta distinção permitirá ao programa interpretar *pixels* contíguos e agrupá-los em regiões. Os algoritmos de segmentação de imagens são baseados em uma ou duas propriedades básicas de valores de níveis de cinza: descontinuidade e similaridade (MASCARENHAS; VELASCO, 1989).

Na descontinuidade, a abordagem é particionar a imagem baseada em mudanças bruscas nos níveis de cinza. As principais áreas de interesse são a detecção de pontos isolados, detecção de linhas e bordas na imagem. Na similaridade, as principais abordagens baseiam-se em limiarização e crescimento de regiões (PRATT, 1991). A próxima seção apresenta os princípios e fundamentos que envolvem estas principais propriedades para as principais técnicas de segmentação de imagem.

### 2.2.1 Detecção de Bordas

A detecção de bordas é um carácter fundamental em processamento digital de imagens, pois contém informações úteis sobre os limites dos objetos que são utilizados na análise e na detecção de objeto no reconhecimento de feições. Também é utilizada para simplificar a análise da imagem, realizando uma redução na quantidade de dados a serem processados, enquanto que simultaneamente, preserva as informações estruturais em torno dos limites do objeto (MEDINA; BELLERA, 2004).

As bordas caracterizam as fronteiras dos objetos e, portanto, são importantes na segmentação dos objetos em cena. As bordas dos objetos em uma imagem são definidas como variações locais dos valores de nível de cinza da imagem, portanto, pode se desenvolver um detector de bordas através de técnicas de diferenciação de imagem, ou seja, uma borda em uma imagem representa uma mudança na intensidade de níveis de cinza nela presente. A transição do nível escuro para brilhante, ou vice-versa, determina uma borda. A boa definição de uma borda em uma imagem depende da fonte de radiação, da iluminação e da distância a que se encontra o objeto da fonte de radiação. Neste caso, é necessário aplicar às imagens operações de filtragem que realçam as mudanças nos valores de cinza e atenuam as áreas da imagem onde existam valores de cinza constantes para, posteriormente, introduzir o resultado desta operação a um detector de bordas por limiar.

Os pontos de contorno são como zonas de *pixels* onde há uma mudança brusca de nível de cinza. No caso de uma imagem com uma função contínua  $f(x, y)$ , percebe-se que sua derivada tem um máximo local na direção do contorno. Portanto, as técnicas mais utilizadas na detecção de bordas são baseadas na medida do gradiente de  $f$  com os gradientes ortogonais  $f_x$  e  $f_y$  ao longo de  $r$  (raio) em uma direção  $\theta$  (Figura 8).

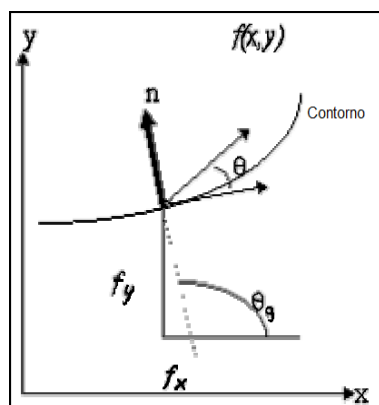


Figura 8. Gradiente de  $f(x, y)$  (GONZALEZ; WINTZ, 2002).

Em geral, utiliza-se o operador gradiente para a detecção descontínua local da luminosidade de uma imagem. A seguir, pode ser observado o comportamento do método de realce de limiar através de operadores diferenciais de 1ª ordem.

A primeira derivada em qualquer ponto da imagem é obtida pela utilização da magnitude do gradiente naquele ponto. A segunda derivada é similarmente obtida através da transformada de Laplace. O gradiente de uma imagem  $f(x,y)$  numa posição  $(x,y)$  é o vetor:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (7)$$

O vetor gradiente aponta na direção da taxa de mudança máxima de  $f$  em  $(x,y)$ . Na detecção de bordas, uma importante quantidade é a magnitude deste vetor, simplesmente chamada de gradiente e denotado por:

$$|\nabla f| = \text{mag}(\nabla f) = \sqrt{G_x^2 + G_y^2} \quad (8)$$

É prática comum aproximar o gradiente através de valores absolutos:

$$|\nabla f| \approx |G_x| + |G_y| \quad (9)$$

A direção do vetor de gradiente também é uma quantidade importante. Seja  $\theta(x,y)$  a direção do ângulo do vetor  $\nabla f$  no ponto  $(x,y)$ , assim da análise vetorial, tem-se:

$$\theta(x,y) = \text{tg}^{-1} \left( \frac{G_y}{G_x} \right) \quad (10)$$

A computação do gradiente de uma imagem é baseado na obtenção das derivadas parciais em cada posição da imagem. Estas derivadas podem ser implementadas no domínio digital de uma variedade de formas.

### 2.2.2 Limiarização (*Thresholding*)

Uma das abordagens para a segmentação de imagens mais importantes é a limiarização (*thresholding*). É um método simples de definir as faixas valores de níveis de cinza da imagem, que consiste em eleger os *pixels* pertencentes ou não a essas faixas: torna-se *pixel* da faixa aquele que a pertence e rejeita-se todos os outros. Como resultado a imagem se

mostra como uma imagem binária (de dois níveis), utilizando as cores branca e preta para distinguir as regiões (não há nenhuma convenção padrão sobre quais são as características de interesse, sejam brancos ou pretos, assim a eleição varia em cada caso). Este tipo de operação é denominado *thresholding* (PEARSON, 1991).

Para elucidar este método, suponha-se que o histograma de níveis de cinza da figura 9(a) corresponde a uma imagem  $f(x, y)$  composta por objetos brilhantes sobre um fundo escuro de tal forma que os *pixels* dos objetos e fundo são meios de seleção. Uma forma lógica de extrair os objetos do fundo é selecionar um limiar  $T$  que separe estes meios. Assim, para qualquer ponto  $(x, y)$ , se  $f(x, y) > T$ ,  $(x, y)$  é chamado de ponto do objeto; senão  $(x, y)$  é chamado ponto de fundo. A figura 9(b) mostra um caso generalizado, no qual existem três meios dominantes de caracterizar o histograma da imagem (por exemplo, dois tipos de objetos brilhantes em um fundo escuro). A mesma aproximação classifica um ponto  $(x, y)$  pertencente a uma classe de objeto se  $T_1 < f(x, y) \leq T_2$ , e a outra classe se  $f(x, y) \geq T_2$ , e do fundo, em caso de  $f(x, y) \leq T_1$ . O *thresholding* multinível é menos confiável do que o simples, visto que é difícil estabelecer vários limiares que isolem as regiões de interesse, especialmente quando o número de meios correspondentes no histograma é grande. Tipicamente, é eficaz utilizar o *thresholding* simples variando-se o nível de limiar (GONZALEZ; WINTZ, 2002).

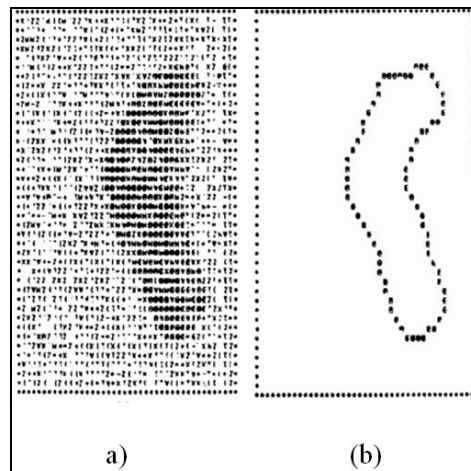


Figura 9. (a) Imagem ruidosa; (b) Resultado da detecção de contornos (GONZALEZ; WINTZ, 2002).

Com base no exposto, *thresholding* pode ser considerado como uma operação que envolve testes em uma função  $T$  de forma que:

$$T = T [ x, y, p (x, y), f (x, y) ] \quad (11)$$

Onde  $f(x, y)$  é o nível de cinza do ponto  $(x, y)$  e  $p(x, y)$  representa alguma propriedade local na vizinhança deste ponto. Uma imagem  $g(x, y)$  a qual se tem aplicado um limiar de detecção se define como:

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) \geq T \\ 0 & \text{se } f(x, y) < T \end{cases} \quad (12)$$

Os princípios que regem esta técnica são a similaridade entre os *pixels* pertencentes a um objeto e suas diferenças em relação aos demais *pixels*. Portanto, ao aplicar-se um limiar, a imagem de níveis de cinza se transformará em binarizada; rotulando-se 1 (ou outro nível de intensidade) os *pixels* correspondentes ao objeto e 0 aqueles que são do fundo da imagem. Quando  $T$  depende unicamente de  $f(x, y)$ , o limiar denomina-se global, mas se depende de  $f(x, y)$ , o limiar denomina-se local. Se,  $T$  depende das coordenadas espaciais  $x$  e  $y$ , o limiar denomina-se dinâmico (GONZALEZ; WINTZ, 2002). Uma das dificuldades da limiarização de uma imagem é a melhor determinação do ponto de separação dos *pixels* (*thresholding*). Para isso existem diversos métodos, os mais utilizados, por serem simples, são o método do vale e o método global simples (*global thresholding*).

A seleção do valor do limiar é definida, geralmente, a partir do histograma da imagem. Assim se uma imagem é composta de um objeto que aparece na cena sobre um fundo, então o histograma é bimodal. Por exemplo, se o objeto é mais claro do que o fundo, no histograma aparecerá dois picos, um localizado nos valores de cinza mais elevados correspondentes ao objeto da imagem, e outro nos valores de cinza mais baixos correspondentes ao fundo da imagem. A Figura 10 mostra um histograma bimodal, no qual o limiar está localizado entre dois picos do histograma.

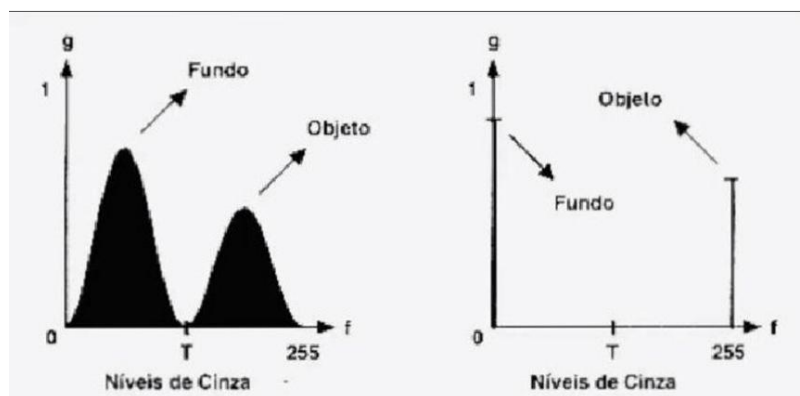


Figura 10. Detecção do fundo e do objeto na imagem (MOIGNE; TILTON, 1995).

Como citado anteriormente, o êxito deste método depende da partição do histograma. A desvantagem é que a binarização (com base no histograma) pode apresentar problemas correspondentes a variações nos tons de cinza originadas por sombras ou devido à iluminação.

### 2.2.3 Crescimento de Regiões

O método de segmentação baseado em regiões atribui um vetor característico a cada região, e segmentam a imagem de acordo com os valores de funções discriminantes adequadas definidas sobre o espaço destas características (MEDINA; BELLERA, 2004). Os procedimentos mais utilizados são o crescimento de regiões e divisão e agrupamento de regiões, os quais são inicializados por *pixels* sementes. Suponha que *pixels* adjacentes têm propriedades semelhantes. Partindo-se do *pixel* semente e aplicando certas condições de inclusão, as regiões crescem e novos *pixels* são admitidos na região (FAN, J. et al. , 2001). O crescimento de regiões é um dos métodos conceitualmente mais simples para a segmentação; os *pixels* adjacentes de amplitude similar se agrupam para formar uma região segmentada. Contudo, na prática, existem limitações, algumas das quais são razoavelmente simples e devem ser levadas em conta para obter um padrão de crescimento que ofereça resultados aceitáveis. A dificuldade desta técnica é a escolha da semente, e a determinação da condição de parada para obter a melhor região. Em geral, os algoritmos de crescimento de regiões fixam essas condições, fazendo com que o algoritmo seja útil para determinados tipos de imagens, mas não para aplicação em geral.

Brice e Fenema (1970) desenvolveram um método de crescimento de regiões baseado em um jogo de regras de crescimento simples. Na primeira parte do processo, combina-se pares de *pixels* quantificados em grupos denominados regiões atômicas (se os *pixels* forem da mesma amplitude) e regiões de conectividade 4. À continuação se aplicam duas regras heurísticas para dissolver as vizinhanças fracas entre vizinhanças atômicas. Fazendo referência à Figura 11, observa-se  $R_1$  e  $R_2$  adjacentes com perímetros  $P_1$  e  $P_2$  respectivamente, perímetros que foram previamente unidos. Após realizar as etapas iniciais para o crescimento de regiões, uma região deve conter previamente sub-regiões unidas por diferentes valores de amplitude ( e. g.  $Y_A$ ,  $Y_B$ ,  $Y_C$ ,  $Y_D$  são sub-regiões da região  $R_1$ ).

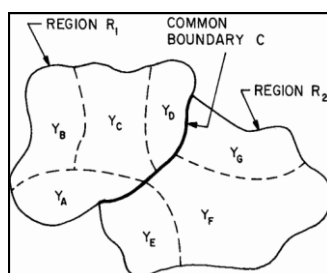


Figura 11. Geometria do crescimento de regiões (PRATT, 1991).



Considerando  $C$  como o tamanho da vizinhança comum e  $D$  o tamanho da porção de  $C$  em que a diferença de amplitude  $Y$  através da região é menor o fator de significância, desta forma duas regiões são ligadas se:

$$\frac{D}{\min \{P_1, P_2\}} \triangleright \varepsilon \quad (13)$$

O método de crescimento de Brice e Fenema (1970) oferece bons resultados para segmentação de imagens com pouca textura.

Uma região pode ser definida como sendo um conjunto de pontos que respeitam um mesmo predicado de homogeneidade. Assim, deve sempre existir pelo menos um caminho inteiramente contido nessa região, ligando dois pontos. Esse segmentador é utilizado quando existem várias regiões em uma imagem.

A mais simples das abordagens é a agregação de *pixel*, a qual começa a partir de um *pixel* semente e com base na medida de homogeneidade, cada célula semente é comparada com as células vizinhas e duas células vizinhas são agrupadas se o critério de similaridade for satisfeito (Figuras 12(a) e 12(b)). Neste caso, ocorre a formação de uma nova célula, que herda as propriedades das duas células anteriores (Figuras 12(c) e 12(d)). As células continuam a ser agrupadas com suas vizinhas enquanto o critério de similaridade for satisfeito (Figuras 12(e) e 12(f)). Esse processo é feito até contemplar toda a imagem. A desvantagem é que os resultados da segmentação são dependentes da escolha dos *pixels* sementes e um ruído na imagem pode fazer com que as sementes sejam mal agrupadas. Técnicas de redução de ruídos de imagens também podem ser aplicadas para resolver esse problema (ESPINDOLA, 2006).

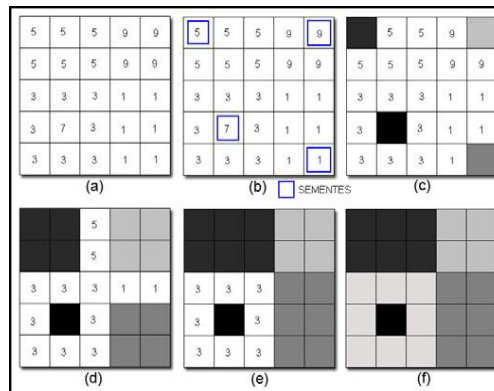


Figura 12. Algoritmo de Segmentação de Imagens do programa *Spring* (ESPINDOLA, 2006).

Este trabalho apresenta um protótipo de um sistema de reconhecimento de feições em imagem implementado através de um sistema multiagente, no qual o seu módulo segmentador de imagem baseia-se nos conceitos da técnica de crescimento de regiões.

## 2.2 Classificação de Imagem Digital

A classificação de imagens digitais é, em certa forma, semelhante à fotointerpretação. Nesta fase, identifica-se o padrão visual associado a cada cobertura (categoria) de acordo com um conjunto de parâmetros tais como o tom, a textura, a forma, o contexto, o *layout*, etc. e depois identifica-se sobre as fotografias as áreas correspondentes a essas categorias por sua semelhança com o padrão previamente identificado. Finalmente, é necessária a verificação dos resultados.

Nas técnicas de classificação de imagens digitais são dadas as mesmas etapas. Desta forma se distingue as seguintes fases (CHUVIECO, 2000):

1. Fase de treinamento: definição digital das categorias.
2. Fase de atribuição: agrupamento dos *pixels* da imagem em uma das categorias previamente definidas.
3. Fase de verificação de resultados.

### 2.3.1 Fase de Treinamento

A classificação digital começa com a definição das categorias que se pretende distinguir na imagem. Trata-se de uma classificação baseada em valores numéricos. No entanto, é necessário obter as faixas de valores de níveis de cinza que identificam cada categoria, que participam da classificação.

As distintas categorias não se definem somente pelas faixas de níveis de cinza, mas também por um conjunto de faixas próximas entre si. Existe certa dispersão em torno da faixa média de cada categoria. Devido a isso a fase de treinamento define com rigor cada uma das categorias que se pretende distinguir tomando em conta sua dispersão na zona de estudo. Isso se consegue selecionando uma amostra de *pixels* na imagem que representa adequadamente cada categoria. A partir disto se determinam valores médios das bandas de cada classe e a faixa em que variam, para as bandas que compõem a classificação. As estimações posteriores se baseiam na amostra selecionada na qual convém selecioná-la adequadamente, já que os resultados da classificação são influenciados pela definição prévia das categorias do que pelo critério pelo qual são diferenciadas na imagem (GÓMEZ; SOTOS, 2010).

Os métodos de classificação podem distinguir-se em dois grupos: método supervisionado e não supervisionado. O método supervisionado parte de um conhecimento prévio da superfície na qual se seleciona as amostras para cada uma das diferentes categorias. Em outras palavras, no método não supervisionado se buscam automaticamente grupos de valores espectrais homogêneos na imagem para que depois o usuário possa encontrar as correspondências entre os grupos selecionados e as categorias que podem ser de interesse.

Neste contexto, é necessário distinguir dois tipos de classes que podem intervir na classificação temática: informativas e espectrais. As classes informativas se estabelecem pelo usuário e formam a legenda de trabalho que se tenta discriminar. As classes espectrais correspondem a grupos de valores espectrais homogêneos (refletividade similar). Estes valores são deduzidos dos níveis de cinza da imagem. O ideal seria produzir uma correspondência perfeita entre somente uma classe informacional (é dizer, uma única classe de cobertura e uma única classe espectral), porém isto é pouco corrente e o comum é que se produza uma das seguintes situações:

- a) Pode ocorrer que uma categoria informacional esteja expressa em várias categorias espectrais. Por exemplo, em uma área urbana poderia aparecer em duas classes espectrais: uma correspondente a sombras e outra a telhados das edificações. Para solucionar este problema define-se corretamente a dispersão espectral de cada classe aperfeiçoando a amostra.
- b) Outro caso seria que duas ou mais classes informativas compartilham uma única classe espectral. Pode-se solucionar unindo as duas categorias em uma generalizada. Por exemplo, pode-se eleger uma classe denominada vegetação em lugar de distinguir entre os distintos tipos de pinhos, ou outros tipos de cultivos, frente a distintas densidades de vegetação. Em caso de não poder unir as duas categorias em uma única, devido a sua heterogeneidade, pode-se analisar uma imagem de outra data na qual a confusão seja evitada.
- c) Pode ocorrer que várias classes informativas compartilham várias classes espectrais. Nestes casos deve-se repensar a estratégia de trabalho ou tentar alguma das soluções comentadas anteriormente.

Uma vez exposto isto, pode-se afirmar que o método de treinamento supervisionado pretende distinguir classes informativas, enquanto que o não supervisionado trata de diferenciar as classes espectrais (Figura 13). Ambos os métodos não são perfeitos, já que no método supervisionado pode-se diferenciar categorias sem um claro significado espectral; e

no método não supervisionado pode-se distinguir classes que não contenham uma informação de interesse.

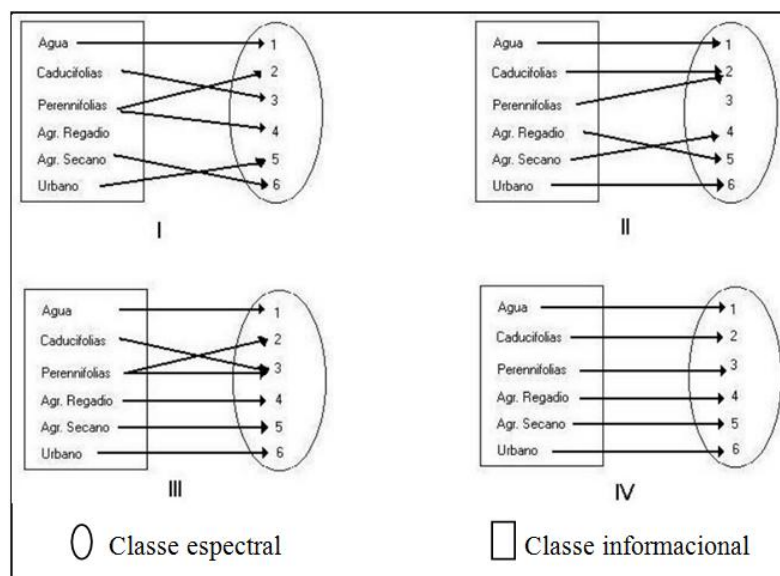


Figura 13. Relacionamento entre classes informativas e espectrais: I uma informacional para muitas espectrais; II uma espectral para muitas informativas; III mescla de ambas; IV relacionamento biunívoco entre elas (GÓMEZ; SOTOS, 2010).

### 2.3.2 Método Supervisionado

Este método de treinamento requer certo conhecimento da zona de estudo que permite delimitar sobre a imagem, zonas ou áreas representativas das categorias que se pretende discriminar. Estas áreas são denominadas como Áreas de Treinamento (*training fields*), que servem para treinar o programa para que possa reconhecer as categorias. A partir destas áreas o programa calcula diversos parâmetros estatísticos dos níveis de cinza que definem cada classe, para assim classificar os *pixels* restantes em uma determinada categoria atendendo a sua faixa de nível de cinza. É apropriado selecionar várias áreas de treinamento por classe para refletir corretamente sua variabilidade na zona de estudo.

Uma vez terminada a seleção das áreas de treinamento, o programa deve proceder calculando as estatísticas elementares de cada categoria: média, desvio padrão, matriz variância e covariância, etc., a partir dos valores de nível de cinza de todos os *pixels* pertencentes à área de treinamento da classe. Este cálculo é aplicado a cada banda espectral. Os valores extraídos a partir dos valores de níveis de cinza de cada área de treinamento

definem as classes, portanto é muito importante sua correta seleção, pois caso contrário o programa classifica áreas heterogêneas (CHUVIECO, 2000).

Para elucidar o método de classificação de imagem supervisionado, suponha-se que em uma imagem se distinguem 3 classes: vinhedo, campos de grãos e de carvalho. A primeira opção no momento de definir as áreas de treinamento seria selecionar uma por categoria. A partir destas áreas de treinamento, o programa calcula uma série de parâmetros que definem uma classe.

A seleção de áreas de treinamento seria incorreta, se alguma destas se situar entre duas ou mais categorias já que ao tratar-se de uma área heterogênea, suas medidas estatísticas seriam uma média entre as distintas classes. Outro erro que se pode ocorrer seria selecionar unicamente como área de treinamento aquelas zonas de cobertura onde esta seja homogênea, já que o desejável seria passar a variação real que tenha esta cobertura. Por exemplo, ao classificar o vinhedo deve-se selecionar como áreas de treinamento tanto zonas onde o vinhedo apareça com uma densidade maior como zonas onde a densidade seja menor.

O tamanho que devem ter as áreas de treinamento requer um tamanho  $m+1$  *pixel* por categoria, sendo  $m$  o número de bandas espectrais que intervêm na classificação. O indicado é eleger várias áreas de pequenos tamanhos ao invés de dimensões maiores. Outros métodos que podem ser utilizados são selecionar *pixels* isolados de cada categoria ou selecionar *pixels* puros em torno dos quais o programa cria uma parcela de modo automático em função da distância ou similaridade espectral (ou ambas) dos *pixels* vizinhos.

### 2.3.3 Método Não Supervisionado

Este método de classificação define as classes espectrais presentes na imagem. Não requer conhecimento prévio da área de estudo, de modo que a intervenção humana se concentra na interpretação dos resultados.

Este método assume que os valores de níveis de cinza da imagem se agrupam em uma série de conglomerados (ou *clusters*) que se correspondem com grupos de *pixels* com um comportamento espectral homogêneo e que definem classes informativas de interesse. Estas classes espectrais não podem ser assimiladas sempre às categorias temáticas que o usuário pretende deduzir pela interpretação do significado temático das categorias espectrais.

O método para definir os grupos com um comportamento espectral similar é parecido com outras técnicas de classificação automática. A seleção baseia-se em três parâmetros: variáveis que intervêm na análise, critério para medir a distância ou similaridade entre *pixels*, e um critério para agrupar os *pixels* similares que compõem a imagem. O foco deste método é encontrar grupos de *pixels* com valores de cinza próximos para atribuí-los às categorias temáticas (FAN, 2001).

A definição de cada grupo espectral se inicia apontando dois critérios: um que marque a similaridade entre os *pixels*, e outro que marque as condições do processo de agrupamento. Para o primeiro, o método mais utilizado se baseia na distância Euclidiana embora possa utilizar a distância média ou de Mahalanobis. O algoritmo de agrupamento mais utilizado é o *k-means* e o *isodata* (FERRERO, 2004).

#### 2.3.4 Fase de Atribuição

Nesta fase são expostos os principais algoritmos de classificação. A partir do ponto de vista estatístico, todos os algoritmos definem uma área de domínio de cada classe em torno de seu centro mediante a um conjunto de funções. Um determinado *pixel* será atribuído a uma classe se seus valores de cinza se encontram dentro dos limites estabelecidos para esta classe.

##### 2.3.4.1 Classificador de Distância Mínima

O critério mais simples para classificar um *pixel* em uma categoria é incluí-lo na mais próxima a ele, ou seja, naquela que minimize a distância entre esses *pixels* e o centróide da classe. A distância não é uma distância geográfica, mas sim espectral resultado da comparação dos valores de cinza de cada *pixel* com os *pixels* centrais de cada categoria, para todas as bandas que participam da análise.

Existem várias formas de medir esta distância espectral entre os *pixels* e classes, uma das mais utilizadas é a distância Euclidiana:

$$d_{ij}^{(k)} = \sqrt{\sum_{n=1}^{N_{bands}} (X_{ij}^n - \bar{X}_{ij}^{(k)})^2}$$

(distância Euclidiana) (12)

Sendo  $X_{ij}^n$  ( $n= 1, 2, 3, \dots, N_{\text{classes}}$ ) o valor de cada *pixel* da banda  $n$ , e  $\bar{X}_{ij}^n$  o valor do centróide (média aritmética) da classe  $k$  na banda  $n$ . Os subíndices  $(i, j)$  correspondem ao número de coluna e linha de cada *pixel* da imagem (CHUVIECO, 2000).

Uma vez determinadas as distâncias de cada classe, seleciona-se a mínima de todas elas, e atribui ao *pixel* correspondente à classe mais próxima (ou seja, a classe para a qual a distância é menor).

$$\text{Classe } (i, j) = \{ k \mid d_{ij}^{(k)} = \text{mínimo} \} \quad (13)$$

Este método é relativamente rápido de executar e oferece bons resultados quando não existe uma sobreposição entre as classes. Além disto, não permite que um *pixel* não seja classificado, visto que existe sempre uma classe mais próxima.

#### 2.3.4.2 Classificador de Paralelepípedo

Neste método, o usuário fixa uma área de domínio para cada categoria considerando seus valores de centralidade e dispersão. Posteriormente cada *pixel* será atribuído a uma classe se seus valores de cinza estão dentro da área de domínio dessa classe, para todas as bandas que participam na classificação. O *pixel*  $x$  será atribuído a classe  $A$  se seus valores de níveis de cinza nas distintas bandas ( $ND_{x,k}$ ) estão incluídos na área de domínio dessa classe:

$$\overline{ND}_{A,n} - R_{A,n} \leq ND_{x,n} \leq \overline{ND}_{A,n} + R_{A,n} \quad (14)$$

Para todo  $n= 1, 2, 3, \dots, N_{\text{bandas}}$ .

$R_{A,n}$  = faixa de dispersão indicada para cada categoria  $A$  em cada banda  $n$ .

Na Figura 14, observa-se alguns problemas que ocorrem neste algoritmo. O *pixel* P1 está classificado sem problemas na classe C1. O *pixel* P2 não está enquadrado em nenhuma área de domínio, assim não será classificado. O *pixel* P3 se encontra em uma zona de sobreposição entre os domínios de duas categorias, assim atribui-se aleatoriamente a duas categorias distintas.

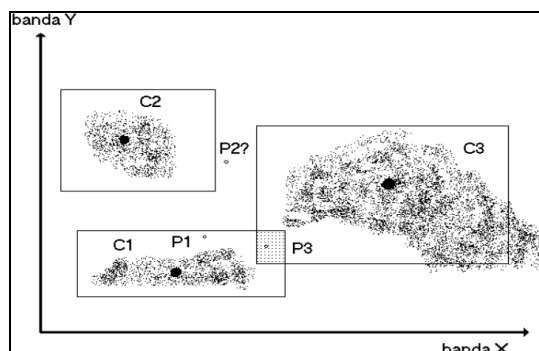


Figura 14. Classificador de paralelepípedos (CHUVIECO, 2000).

Este método é rápido de executar já que necessita somente de operadores condicionais do tipo IF...THEN. Mas apresenta certos problemas inerentes ao desenho das áreas de domínio que podem existir certos *pixels* que não são classificados e outros que se encontram em zonas comuns a duas ou mais categorias. Neste último caso o programa atribuirá este *pixel* a classe que se encontra antes no algoritmo de classificação, embora pode-se utilizar conjuntamente o algoritmo de distância mínima para evitar a sobreposição. A solução para os *pixels* sem classificação é ampliar a faixa de dispersão e revisar a definição espectral das categorias, visto que a presença *pixels* não classificados indicam que as classes que não estão bem definidas ou não incluídas na legenda do trabalho.

#### 2.3.4.3 Classificador de Máxima Verossimilhança

Este método considera que os valores de cinza de cada categoria se ajustam a uma distribuição normal. Isto permite descrever essa categoria por uma função de probabilidade, a partir de seu vetor de médias e matriz variância-covariância. Assim calcula-se a probabilidade de que um determinado *pixel* pertença a uma categoria; realiza-se o cálculo para todas as categorias definidas e o *pixel* é atribuído à classe a qual tenha a maior probabilidade de pertencer. A classificação do *pixel* não classificado ocorre conforme um critério de probabilidade Gaussiana. Este classificador é o mais complexo e, portanto, o que maior volume de cálculo requer, porém é o mais robusto e confiável, pois se ajusta fielmente à distribuição original dos dados (RICHARDS; JIA, 1999).



Para calcular a probabilidade associada a certo valor de cinza são necessários os valores das médias e desvio padrão de cada categoria. No caso de trabalhar com uma única banda, a probabilidade é calculada da seguinte forma:

$$p(x/A) = \frac{1}{\sqrt{2\pi\sigma_A^2}} e^{\left\{-(ND_x - \overline{ND_A})^2 / 2\sigma_A^2\right\}} \quad (15)$$

$P(x/A)$  = probabilidade de que um *pixel*  $x$  (definido pelo  $ND_x$ ) pertença à classe  $A$ ,

$\overline{ND_A}$  = média da classe  $A$ ,

$\sigma_A^2$  = variância da classe  $A$ .

Uma vez determinada a probabilidade para todas as categorias o *pixel* será atribuído a classe  $A$ , somente se:

$$p(x/A) \geq p(x/B) \quad (16)$$

$A \neq B$ , com  $B = 1, 2, 3, \dots, m$ ;  $m$  = número de categorias.

Para o caso de duas bandas, os limites das categorias têm uma forma elipsoidal (Figura 15). Cada uma destas elipses é uma linha de isoprobabilidade, ou seja, todos seus pontos têm a mesma probabilidade de pertencer à categoria associada.

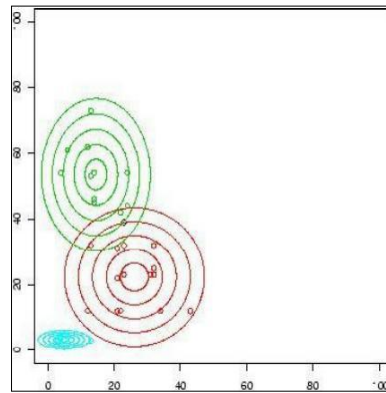


Figura 15. Extensão a duas bandas do algoritmo de máxima verossimilhança.

Com mais de duas bandas é impossível representar graficamente este critério de classificação e deve-se ampliar a fórmula da probabilidade. Uma vez que o programa tenha feito essa atribuição, os *pixels* podem estabelecer um critério que permite atribuí-los a uma classe única se a probabilidade correspondente seja superior a um limiar. Este critério assume que as categorias apresentam uma distribuição normal; contudo, a hipótese de que os dados de refletividade continuam com uma distribuição normal não se cumpre sempre e deve ser verificada.

#### 2.3.4.4 Classificador de Mahalanobis

A distância de Mahalanobis é definida como (RICHARDS; JIA, 1999):

$$d(x, m_i) = \sqrt{(x - m_i)^t \cdot \Sigma^{-1} \cdot (x - m_i)} \quad (17)$$

Onde,  $m_i$  é o vetor médio e  $\Sigma$  a matriz de covariância das amostras.

Este procedimento ocorre quando um vetor  $x$  pertence à  $i$ -ésima classe se a distância entre ele e o centróide da  $i$ -ésima classe é mínima. Esta distância é comumente utilizada para diferentes propósitos estatísticos, por exemplo, para a detecção de dados fora da classificação (*outliers*).

#### 2.3.4.5 Outros Critérios de Classificação

Segue uma descrição sucinta de outros classificadores:

- Classificador em árvore: trata-se de uma análise exaustiva das respostas espectrais das classes e do conjunto de dados disponíveis. Consiste em fazer perguntas a cada *pixel* cuja resposta positiva ou negativa conduzirá a outra pergunta e assim sucessivamente até obter a classe a que pertence. Baseia-se nos mesmos princípios que os sistemas inteligentes.
- Classificador de contexto espacial: todos os métodos vistos utilizam somente valores de refletividade coletados do *pixel* a ser classificado. Isto não é problema se o tamanho do *pixel* for maior do que os diferentes elementos físicos que compõem a paisagem. No entanto, se o tamanho do *pixel* é menor do que a unidade da paisagem pode-se utilizar as informações dos *pixels* adjacentes para estimar a associação de classe ou confirmar a estimativa. Este pode ser um método útil em combinação com a máxima verossimilhança e para incluir informações de *pixels* vizinhos para tomar uma decisão em casos de as probabilidades para as duas classes sejam semelhantes ou exista uma classe com probabilidade de pertinência suficientemente alta.

Este método envolve outras fontes de informação além das bandas para a classificação. Entre essas informações estão a altitude, a declividade, a litologia etc. As diferentes informações sobre a vegetação tem maior capacidade para desenvolver-se em determinados ambientes definidos em parte por essas variáveis. Portanto, é preciso estabelecer qual é o tipo de vegetação com maior probabilidade de se desenvolver em um determinado

*pixel* (dependendo da topografia e litologia) e combinar estas informações com as informações das bandas.

– Classificadores baseados em redes neurais: baseiam-se na utilização de redes neurais artificiais que, supõem-se, imitam as redes neurais reais no desenvolvimento das tarefas de aprendizagem. Um neurônio artificial é um objeto lógico (trata-se de *programa* e não de hardware) que recebe diversas entradas, faz uma soma ponderada das mesmas e produz uma saída a partir da aplicação de uma função de limiar com a da média ponderada.

Se as saídas de um neurônio são conectadas como entradas de outro, obtém-se uma rede neural. Um dos exemplos mais típicos de uma rede neural é a *Back Propagation Neural Network* (Rede Neural de Retroprojeção), como mostra a Figura 16.

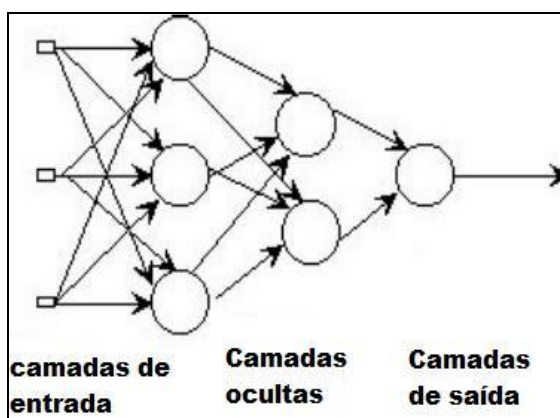


Figura 16. Rede neural *Back Propagation Neural Network* (RICHARDS, 1994).

Este método consta de uma camada de entrada com tantos neurônios como variáveis de entrada que se introduz no modelo (no caso de sensoriamento remoto é considerada uma por cada banda utilizada para classificação), uma oculta que realiza a maior parte do cálculo e uma camada de saída com neurônios e as classes existentes. Em sensoriamento remoto esta saída, geralmente, consiste em um valor numérico entre 0 e 1 para cada classe, quanto maior for este valor maior a probabilidade do *pixel* pertencer a classe em questão (RICHARDS, 1994).

Para trabalhar com uma rede neural é preciso cumprir algumas etapas:

– Treinamento: Introduce-se a rede a resposta espectral dos *pixels* cuja classe se conhece e compara-se a saída com a realidade. A partir desta comparação modificam-se os coeficientes de ponderação de todos os neurônios para que se obtenha a resposta adequada (trata-se de um procedimento automático), ou seja, 1 na classe correta e 0 na incorreta.

- Estabilização: No início do treinamento, os fatores de ponderação mudam rapidamente, mas conforme este se desenvolve (e se as áreas de treinamento forem selecionadas corretamente) se estabilizam (não se modificam ainda que voltem a introduzir *pixel* de treinamento). Neste momento finaliza-se a fase de treinamento.

- Classificação: introduz-se valores espectrais dos *pixels* cujas respostas da classe não se conhece e atribui-se a classe que fornece uma resposta de valor alto.

Trata-se em definitivo de um método de classificação robusto que oferece bons resultados quando as respostas espectrais das classes seguem uma distribuição normal. A chave está no conjunto de coeficientes de ponderação, que constituem um conjunto de parâmetros que devem ajustar-se aos dados de entrada e saída.

### 2.3.5 Fase de Verificação

Nenhuma classificação pode considerar-se completa até que seu grau de exatidão seja avaliado. Este pode ser definido como o grau de concordância entre as classes definidas pelo classificador e suas corretas posições segundo os dados coletados pelo analista e considerados como dados de referência.

O instrumento mais utilizado para avaliar a exatidão de uma classificação é a matriz de confusão, também chamada de matriz de erro ou de contingência. Esta é uma matriz quadrada de  $n \times n$ , onde  $n$  é o número de classes. Esta matriz mostra a relação entre as séries de medidas correspondentes à área de estudo. A primeira série corresponde aos dados de referência adquiridos de observações no campo, inspeções de estatísticas agrícolas, interpretações de fotos aéreas e outras fontes similares. A segunda corresponde à categorização dos *pixels* realizada pelo classificador para as classes de interesse. Em uma matriz de confusão as colunas correspondem aos dados de referência, enquanto que as linhas correspondem às atribuições do classificador.

A partir de uma matriz de confusão pode-se deduzir vários índices relativos à exatidão da classificação. Através da Tabela de valores da matriz confusão (Tabela 1) é possível definir e comentar estes índices.

Tabela 1 - Exemplo de matriz de confusão

Categoria deduzida	Classes de referência			Total
	A	B	C	
A	35	2	2	39
B	10	37	3	50
C	5	1	41	47
Total	50	40	46	136

Os elementos da diagonal indicam os *pixels* que foram corretamente classificados. Todos os elementos fora da diagonal indicam erros de omissão ou comissão. Os erros de omissão correspondem aos elementos não diagonais das colunas (exemplo, um total de 15 *pixels* que deveriam ter sido classificados como classe A, foram atribuídos a outra classe). Os erros de comissão correspondem aos elementos não diagonais nas linhas (exemplo, 5 *pixels* da classe A e 1 da classe B foram incluídos erroneamente na classe C).

A partir da matriz confusão pode-se calcular várias medidas de interesse do ponto de vista da exatidão:

- Exatidão Global: Calcula-se dividindo o número total de *pixels* corretamente classificados pelo número total de *pixels* de referência. Este índice é expresso como porcentagem na Tabela 1:

$$((35 + 37 + 41) / 136) * 100 = 83,1\%$$

Este índice é global e não indica como a exatidão se divide entre as diversas categorias. Muitas vezes as categorias apresentam exatidões diferentes, mas que se combinam de modo que isto não reflita na exatidão global. Examinando-se a matriz confusão é possível utilizar dois métodos para avaliar a exatidão das categorias individuais:

- Exatidão do usuário: Calcula-se dividindo o número dos *pixels* corretamente classificados em cada categoria pelo número total dos *pixels* que foram classificados em determinada categoria (total de linhas). Para a classe A da Tabela 1, tem-se:

$$(35 / 39) * 100 = 89,7\%$$

Este índice se justifica no caso do usuário estar especialmente interessado na porcentagem de cada classe que foi corretamente classificada.

– Exatidão do produtor: Resulta em dividir o número de *pixels* corretamente classificados em cada categoria pelo número de *pixels* de referência para uma categoria (total de coluna). Neste caso temos para a classe A da Tabela 1:

$$(35/50) * 100 = 70,0\%$$

As medidas de exatidão que foram descritas anteriormente, ainda que simples, são baseadas apenas na diagonal principal e colunas da matriz de confusão. Estas medidas não fornecem informações contidas na matriz completa. Pode também ocorrer uma distribuição aleatória dos *pixels* nas classes que podem conduzir a resultados aparentemente corretos na matriz de confusão (RICHARDS; JIA, 1999).

Para eliminar estes inconvenientes foi proposto um índice denominado coeficiente Kappa. É uma medida da diferença entre a exatidão alcançada na classificação com um classificador automático e a chance de alcançar uma classificação correta com um classificador aleatório. O coeficiente Kappa ( $k$ ) é expresso como:

$$\kappa = \frac{N \sum_{i=1}^r x_{(i,i)} - \sum_{i=1}^r (x_{(i,+)} x_{(+,i)})}{N^2 - \sum_{i=1}^r (x_{(i,+)} x_{(+,i)})} \quad (18)$$

Onde  $r$  é o número de linhas da matriz,  $x_{(i,i)}$  indica o observado,  $x_{(i,+)}$  é o total de observações na linha  $i$ ,  $x_{(+,i)}$  é o total de observações na coluna  $i$  e  $N$  é o total de observações incluídas na matriz.

O valor de  $k$  igual a 1 indica um acordo pleno entre a realidade e a classificação, enquanto o valor acerca de 0 sugere que a observação está classificada errada. No caso de valores negativos, pressupõe-se também uma classificação ruim, resultado que pode ser de outros fatores externos, como o próprio tamanho da amostra (CHUVIECO, 2000).

### 2.3 Abordagem Proposta

Conforme visto anteriormente, a segmentação permite distinguir os objetos de interesse de uma imagem, isolando-os do fundo ou de outros objetos. Após a segmentação de uma imagem, cada *pixel* é atribuído a uma região e cada região corresponde a uma parte

semântica da imagem. As discontinuidades entre as regiões correspondem às bordas dos objetos.

As abordagens sobre segmentação podem ser divididas em dois grupos. Por um lado, as abordagens tratam dos métodos para extrair os contornos presentes na imagem, com base no estudo de mudanças bruscas em função da intensidade de níveis de cinza. Por outro lado, aborda o estudo das regiões para detectar as áreas da imagem, que mostram características de homogeneidade e satisfaz um critério de homogeneidade e de semelhança.

Apesar da multiplicidade de abordagens, ainda é difícil identificar a homogeneidade dentro da imagem, tais como a definição de um limiar, ou uma segmentação convencional, como o crescimento por regiões, pois essa homogeneidade pode ter graus variáveis e em diferentes partes da imagem.

Um estudo adaptativo é necessário. Com esse enfoque, será permitido aplicar diferentes operadores em diferentes regiões, e tratar as mudanças específicas relacionadas à intensidade, a nível local dentro de uma mesma região.

Uma estrutura para esta adaptabilidade é proporcionada através dos métodos baseados na resolução coletiva de problemas. Através da integração de uma arquitetura multiagente, pode-se obter essa capacidade, visto que um agente é capaz de controlar seu comportamento e as habilidades de comunicação que o permite ter adaptabilidade. A especialização e a distribuição dos agentes também contribuem para a adaptabilidade do sistema.

Este trabalho tem como objetivo propor uma abordagem para a segmentação e reconhecimento da feição edificação em imagens baseada em arquitetura multiagente.

A próxima seção apresenta as características, tipos e funcionalidades de um agente inteligente que integra um sistema multiagente capaz de realizar as tarefas de segmentação e classificação de imagem, com o objetivo de reconhecer as feições das edificações nos diferentes cenários que compõe uma imagem aérea.

## 2.4 Agentes Inteligentes

Na década de 80 os sistemas de Inteligência Artificial tiveram um progresso onde seus sistemas puderam ter um domínio mais dinâmico. Ao invés de focar resultados simulados, simbólicos em mundos artificiais, começaram a explorar as possibilidades de interações complexas com o mundo físico, através de um mecanismo denominado agente inteligente.

A definição de agente Inteligente ainda é assunto em debate e origina muita discussão sobre o que são e de como eles diferem de um programa em geral. Verificou-se que diferentes pesquisadores propõem diversas definições, que vão do nível elementar até o nível mais elaborado. Sendo assim, várias são as definições correntes:

“Um agente é uma entidade autônoma que tem conhecimentos especializados, possui uma representação desses conhecimentos, possui uma representação do problema a tratar, possui uma representação parcial do ambiente, possui protocolos de comunicação especializados e às vezes um modelo de suas intenções.” (SCALABRIN, 96).

“Um agente é um sistema de computador que está situado em algum ambiente e que é capaz de executar ações autônomas de forma flexível neste ambiente, a fim de satisfazer seus objetivos de projeto.” (WOOLDRIDGE, 1999).

“Um agente é uma entidade que percebe seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores.” (RUSSEL; NORVIG, 2003).

A Figura 17 ilustra um agente inteligente que atua em um ambiente e é capaz de tomar ações autônomas de acordo com o ambiente para cumprir seus objetivos. As percepções são captadas pelos sensores. Os atores executam uma ação que transformam o ambiente. As ações são racionais e de acordo com as percepções captadas.

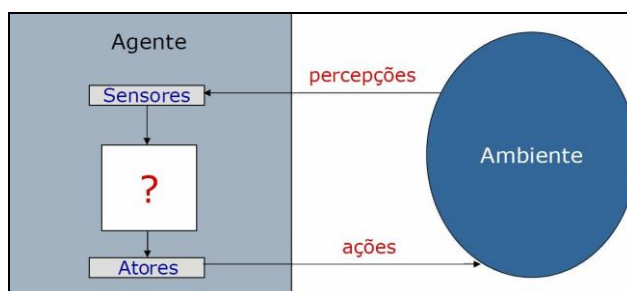


Figura 17. Visão geral de agente inteligente (DAVIDSSON, 1992).

Um agente é um nó processador que deve possuir:

- Autonomia para raciocinar e decidir;
- Ser capaz de interagir com outros agentes, sistemas ou usuários;
- Ter conhecimento e independência para solucionar um problema.

Existem três conceitos comuns nestas definições: ambiente, autonomia e flexibilidade. Estar situado em um ambiente corresponde à capacidade do agente de receber entrada através de sensores do ambiente e executar ações que alterem o mesmo de alguma maneira através de reagentes. A palavra autonomia, neste caso, significa que cada agente possui sua própria existência, a qual não é dependente da existência de outros agentes.

Agente é um conceito de desenvolvimento de programa e para que este seja diferenciado dos demais programas ele tem que ter uma série de características. Uma série de



qualificadores são características, que vêm para designar certas propriedades executadas pelos agentes. Um agente inteligente é um agente que deve ter o comportamento autônomo e flexível, onde a flexibilidade significa que o agente deve ser reativo, pró-ativo e ter habilidades sociais. Wooldridge e Jennings (1995) propõem outra definição para agente bastante aceitável que usa três qualificadores, que são considerados básicos.

- Reagente: o agente é capaz de responder as mudanças do ambiente em que ele está localizado;
- Pró-ativo: o agente deve ser capaz de tentar cumprir as suas próprias tarefas ou objetivos;
- Habilidade social: o agente deve ser capaz de se comunicar com outros agentes utilizando algum tipo de linguagem de comunicação do agente.

### 2.5.1 Características dos Agentes Inteligentes

Conforme Franklin e Graesser (1996), algumas características são atribuídas em algum grau para resolver problemas particulares dos agentes. O comportamento dos agentes dentro de um ambiente é definido por suas características em menor ou maior grau para resolver problemas específicos como:

- Adaptabilidade: refere-se à aprendizagem que um agente é capaz de realizar, e na possibilidade de mudar seu comportamento com base na aprendizagem;
- Autonomia: um agente é completamente autônomo se é capaz de atuar no ambiente baseado no seu aprendizado de forma independente. O agente é capaz de adaptar-se a um ambiente se este muda severamente;
- Benevolência: é a suposição de que um agente está disposto a ajudar os outros agentes, se não entrarem em conflito com seus próprios objetivos;
- Colaboração: o agente deve considerar informações relevantes sobre o usuário antes de executar uma tarefa. Um agente deve verificar possíveis erros humanos por conta dos usuários. Deve ser permitido a um agente recusar ou executar tarefas que possam causar danos ao sistema;
- Comunicabilidade: este atributo permite que um agente possa se comunicar com outros agentes ou com outras entidades.

- Degradação Gradual: é a capacidade do agente executar parte de uma tarefa quando existe incompatibilidade na comunicação ou domínio. No contexto das noções de risco, agentes trabalham melhor quando apresentam esta característica;
- Flexibilidade: um agente deve ser capaz de fazer uma escolha dinâmica das ações e da sequência de execução das mesmas, em um determinado estado do ambiente;
- Inteligência: o raciocínio é estabelecido por meio do conhecimento, das regras e da evolução artificial. O agente sempre executa a ação correta a partir das informações que percebe do ambiente;
- Mobilidade: é a habilidade de um agente se mover pela rede telemática;
- Planejamento: é a capacidade de direcionar as ações para alcançar os objetivos desejados;
- Pró-Atividade: um agente é pró-ativo quando ele é capaz de controlar seus próprios objetivos, apesar das mudanças no ambiente;
- Reatividade: um agente atua conforme as mudanças em seu ambiente ocorrem. Neste caso, um agente percebe o ambiente e essas mudanças direcionam o seu comportamento;
- Representatividade: as ações de um agente devem representar o usuário.

### 2.5.2 Tipos de Agentes Inteligentes

O conjunto de atributos apresentados na seção anterior gerou uma classificação para os diversos tipos de agentes, de acordo com diferentes combinações desses atributos. A classificação tradicional considera três tipos de agentes (JENNINGS, 1995):

#### Agente Reativo

Os agentes reativos reagem a mudanças em seu ambiente, de outros agentes ou de mensagens. Tais mudanças e mensagens podem provocar ativação das ações ou tarefas que o agente reativo possui. Por sua vez, um agente reativo não é capaz de raciocinar sobre seus desejos ou objetivos.

### Agente Deliberativo

Eles são agentes capazes de raciocinar a partir de intenções (objetivos do agente) e crenças (conhecimento considerado pelo agente como verdadeiro), e de criar e executar planos de ação a seguir. Eles podem selecionar um dos seus objetivos possíveis de acordo com suas motivações, detectar conflitos e coincidências na criação dos planos de ação, etc.

### Agente Híbrido

Agentes híbridos possuem características de agentes deliberativos e reativos simultaneamente. O objetivo é organizar as funcionalidades do agente hierarquicamente em camadas onde, geralmente, a camada reativa tem prioridade sobre a deliberativa, com a finalidade de obter uma resposta rápida.

#### 2.5.3 Arquitetura dos Agentes Inteligentes

Um agente percebe o ambiente e realiza as ações para alcançar de forma autônoma os seus objetivos. O comportamento do agente é descrito pela função agente que mapeia dados das percepções para as ações. O que é programado no agente são os objetivos e os mecanismos que esses objetivos são realizados, a sequência de ações (DAVIDSSON, 1992).

Uma arquitetura para agentes permite decompor um sistema de agentes e determinar como devem ser as interações entre eles e com o ambiente. Distinguem-se algumas arquiteturas que se classificam segundo o modelo de raciocínio que utilizam (Figura 18).

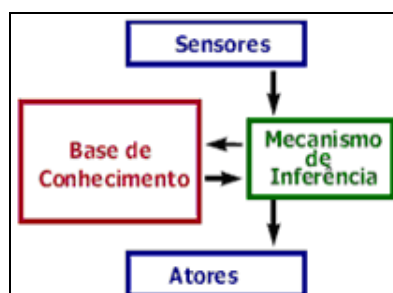


Figura 18. Arquitetura básica de um agente autônomo (DAVIDSSON, 1992).

## Agentes Reativos Simples

O agente utiliza regras de condição-ação para estabelecer a conexão entre as percepções e ações. O agente atua procurando uma regra cuja condição coincida com a situação corrente, definida pela percepção, e executando a ação correspondente a essa regra.

A Figura 19 mostra como o agente interage com o ambiente com ações e reações simples, tomando como partida a coleta de dados através da utilização de sensores.

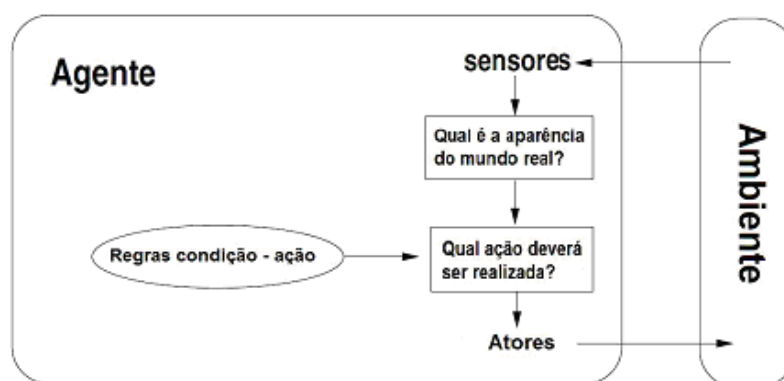


Figura 19. Arquitetura básica de um agente reativo simples (RUSSEL; NORVIG, 2003).

## Agentes Reativos Baseados em Modelo

A Figura 20 estende a Figura 19 acrescentando-se três conceitos, os quais formam a determinação do estado deste agente: a atualização do estado, a evolução do mundo e as ações do agente. O agente mantém um tipo de estado interno, que é atualizado com informações sobre como o mundo evolui independentemente do agente, e de como as ações do agente afetam o mundo.

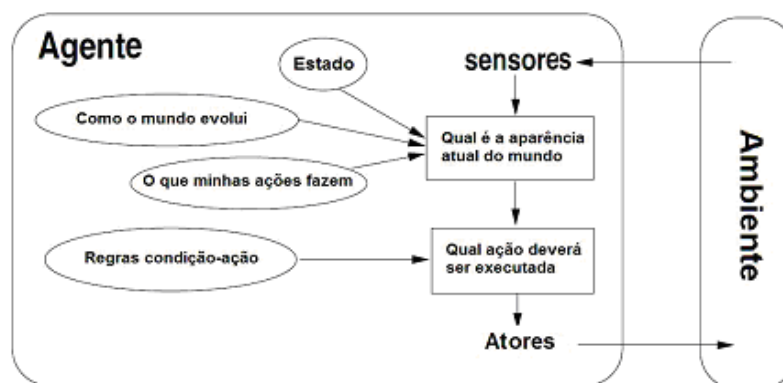


Figura 20. Agente reativo baseado em modelo (RUSSEL; NORVIG, 2003).

### Agentes Reativos Baseados em Objetivo

Para o agente decidir o que fazer há necessidade de informações sobre seus objetivos, e de informações que detalham os seus desejos. O agente tem que escolher as possíveis ações que lhe permitam alcançar seus objetivos.

A Figura 21 mostra que um agente utiliza as informações obtidas sobre suas metas, e também utiliza as informações das ações que serão realizadas, ou seja, o planejamento dos resultados que se obterá com as reações que acontecerão no futuro. Deste modo o agente poderá saber qual é melhor maneira de tomar uma decisão.

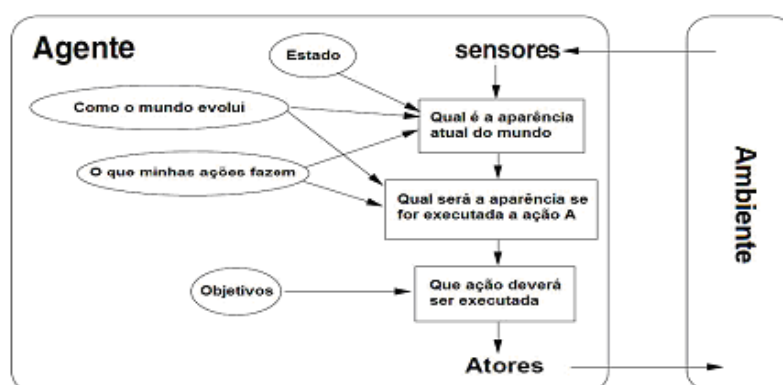


Figura 21. Agente baseado em objetivo (RUSSEL; NORVIG, 2003).

### Agentes Reativos Baseados na Utilidade

Os objetivos não são suficientes para gerar um comportamento adequado, assim, se escolhe um estado do mundo, e este estado então oferece maior utilidade para o agente. A utilidade é uma função que associa o seu grau de utilidade ao estado.

A especificação completa da função de utilidade permite a tomada de decisões racionais quando satisfazem alguns objetivos que envolvem conflito no caso do agente desejar obter vários objetivos e não houver certeza de alcançá-los (SEEL, 1989).

Na Figura 22 o agente utiliza um modelo do ambiente, juntamente com uma função de utilidade que calcula suas preferências entre os estados do ambiente. Em seguida, seleciona a ação que o leva a alcançar a utilidade máxima esperada, entre todos os resultados possíveis.

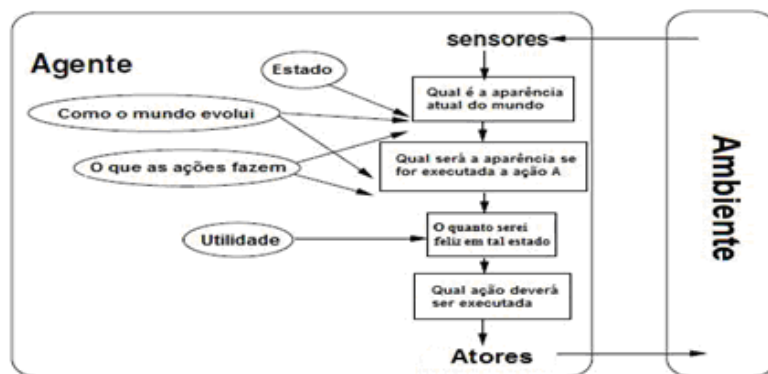


Figura 22. Agente baseado na utilidade (RUSSEL; NORVIG, 2003).

### Agentes com Aprendizagem

Um elemento de aprendizado utiliza realimentação sobre como um agente está funcionando e determina de que maneira o elemento de desempenho deve ser modificado para funcionar melhor no futuro (Figura 23).

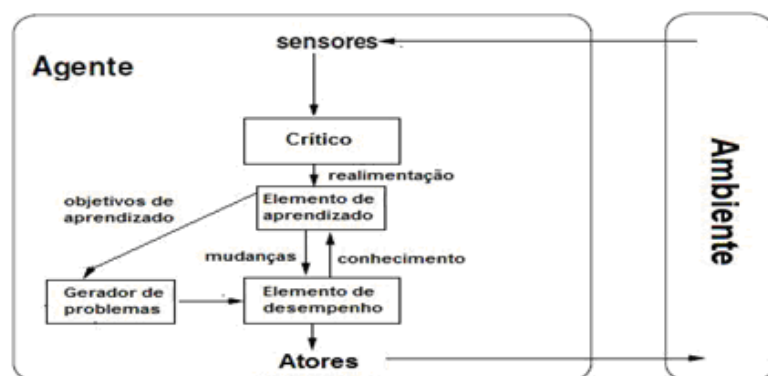


Figura 23. Agente com aprendizagem (RUSSEL; NORVIG, 2003).

#### 2.5.4 Estados Mentais

Os seres humanos costumam usar frases como: "Luisa leva um guarda-chuva porque acredita que vai chover". Esta forma de estrutura de idéia permite aos psicólogos explicarem o comportamento do ser humano ou explicá-lo mediante as atitudes mentais, tais como crenças, desejos, medos, temores e etc. Dennett (1987) fala dos sistemas intencionais que descrevem as entidades cujo comportamento pode ser previsto pelo método de atribuir crenças, desejos e capacidade racional.

Se o sistema é muito simples, podem ser atribuídas descrições intencionais consistentes com o comportamento observado (KAELBLING; ROSENSCHEIN, 1986). No entanto, em sistemas complexos as noções intencionais fornecem as ferramentas de abstração, que proporcionam uma forma adequada e familiar de descrever, explicar e prever o comportamento de sistemas complexos (SEEL, 1989).

Portanto, um agente pode ser visto como um sistema capaz de ser definido adequadamente pela postura intencional. Um agente contém atitudes que proporcionam informação (atitudes de informação) e pró-atitudes.

As atitudes de informações relacionadas com o conhecimento que um agente tem sobre seu ambiente, enquanto pró-atitudes são aquelas que de alguma forma orientam as ações do agente. A escolha de qual combinação de atitudes é a mais adequada para caracterizar os agentes é motivo de grandes discussões. Essa combinação possui, pelo menos, uma atitude de informação e uma pró-atitude (Tabela 2). Um agente toma decisões e possui intenções com base na informação disponível sobre seu ambiente, portanto, a relação entre as duas categorias de atitudes é uma conexão fechada.

Tabela 2 - Atitudes de informação e pró-atitudes de um agente

Atitudes de informação	Pró-atitudes
	Desejo
Crenças	Intenção
Conhecimento	Obrigação
	Compromisso
	Eleição

Não há um consenso entre o mundo da inteligência artificial e da comunidade de filósofos sobre qual é a combinação entre as atitudes de informação e pró-atitudes são mais adequadas para definir um agente. Uma revisão feita por diferentes autores indicou atitudes e pró-atitudes que caracterizam um agente.

Rao e Georgeff (1993) desenvolveram uma estrutura lógica para a teoria do agente com base em três primitivas: crenças, desejos e intenções. Seu formalismo é baseado em uma extensão do "modelo de tempo", em que os mundos acessíveis de crenças, desejos e intenções são eles próprios extensões desse modelo.

Wooldridge (1992) desenvolveu uma família de lógicas para representar propriedades de sistemas multiagentes. Seus formalismos são usados na especificação e verificação desses sistemas. Ele desenvolveu um modelo simples e genérico de sistemas multiagentes realista, e mostrou exemplos de como estas lógicas são usados na especificação e verificação de protocolos de cooperação.

Segundo Rao e Georgeff (1993) um sistema pode ser modelado como um agente com atitudes mentais de crenças (*Beliefs*), desejo (*Desire*) e intenções (*Intentions*). Embora não exista um ponto de vista único nesta área para saber se essas três atitudes são suficientes na abordagem que se apresenta, os autores consideram que, de acordo com sua experiência e razões práticas, essas três atitudes são necessárias e suficientes. Na definição proposta por, cada uma das atitudes considerada no projeto de agentes racionais é então introduzida. Um agente deve ser capaz de agir de forma autônoma em um ambiente em mudança, para isso é necessário que um componente do sistema armazene as informações que tem sobre o ambiente em que o agente está envolvido, de forma que se pode alterá-lo quando os sensores detectam mudanças em seu ambiente. Este componente do sistema é a crença.

O sistema deve ter informações sobre os objetivos a serem alcançados e as prioridades associadas a cada um dos objetivos. Em um agente não existe um único objetivo, como na maioria dos sistemas de Inteligência Artificial, mas vários devem ser priorizados e podem até ser incompatíveis. Estes objetivos formam a parte dos desejos do sistema.

Uma vez que o ambiente muda, pode ser necessário modificar o curso da ação a ser tomada para alcançar um objetivo. Portanto, se inclui um componente que representa o caminho escolhido pelo sistema para alcançar os objetivos. Este componente é chamado de intenção.

Em suma, pode-se dizer que as crenças representam a parte de informação do sistema, os desejos, a parte das motivações e as intenções, a parte e deliberada do sistema.



### 2.5.5 Sistema Multiagente

Na maioria dos casos, os agentes não são desenvolvidos de forma independente, e sim como entidades que constituem um sistema. A este sistema se denomina multiagente (FERBER, 1999).

Neste caso, os agentes devem ou podem interagir entre si. As interações mais habituais são informar ou consultar a outros agentes que constituem o sistema.

Podemos notar pela Figura 24, que cada agente tem uma visão parcial do ambiente onde se quer resolver o problema. Uma solução para o problema é obtida pela criação de um sistema em que, múltiplos componentes autônomos com objetivos particulares se interrelacionem.

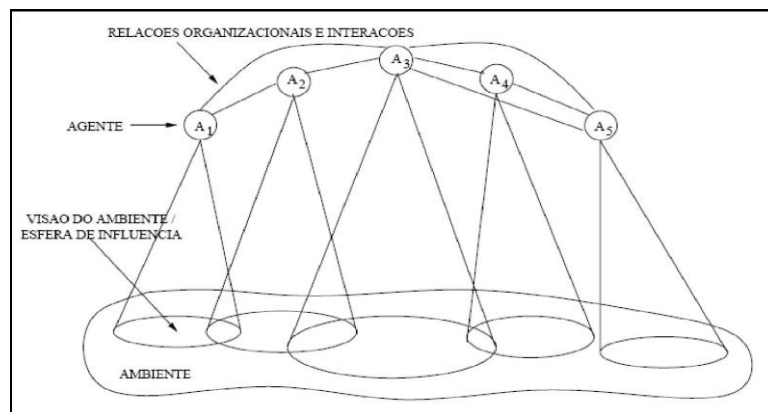


Figura 24. Visão de um Sistema Multiagente (HUBNER; SICHMAN, 2003).

### 2.5.6 Aspectos Fundamentais

Num sistema multiagente as ações de cada agente fazem sentido em relação aos objetivos comuns do grupo, ou seja, os agentes agem de modo a cooperarem ou competirem uns com os outros para atingir um objetivo em comum. A presença de alguns dos aspectos a seguir é de relevante consideração (WOOLDRIDGE, 2002):

- Organização: consiste na atuação coordenada entre os agentes, de modo a colaborar na resolução de uma tarefa;
- Cooperação: trata de como os agentes se comportam individual e socialmente de forma que, os objetivos sejam realizados;

- Negociação: É uma ferramenta importante na interação de sistemas multiagentes, onde promove a coordenação e cooperação entre os agentes. Busca auxiliar os agentes no intercâmbio de informações com o propósito de solucionar objetivos conflitantes;
- Comunicação: realiza a troca de informações entre os agentes.

Os sistemas multiagentes são responsáveis pela especificação de todo o sistema, incluindo a coordenação dos agentes, assim como a comunicação, a coordenação, a colaboração, a divisão de tarefas e resolução de conflitos através da negociação entre agentes.

### 2.5.7 Linguagem de Agentes

A linguagem dos agentes permite programar sistemas computacionais de acordo com alguns conceitos desenvolvidos pela teoria de agentes. Mesmo que exista uma estrutura que represente um agente, deve-se esperar que tal linguagem tenha conceitos baseados na teoria de agentes, tais como crenças, metas, objetivos, planejamento e etc.

O interesse pela linguagem de agentes surgiu da proposta de Shoham (1993) para criar um novo paradigma de programação orientada a agentes. Uma linguagem de programação multiagente proposta foi o Agent0, onde as principais entidades são o agente e o seu estado (um conjunto de crenças, habilidades, ações e objetivos). Existem também métodos para a troca de mensagens, solicitação e distribuição de tarefas.

Outra alternativa são as abordagens orientadas para as teorias dos agentes, que estabelecem definições dos sistemas multiagentes para então tratar estas como implementações. Um exemplo dessa linguagem alternativa é ConCOLOG onde se modela a execução das tarefas distribuídas a vários agentes e como estas modificam o ambiente. A implementação destas características de um sistema multiagente é baseada em entidades de conhecimento modificadas pelas tarefas. O conjunto de aplicações que pode-se criar sobre as tarefas são axiomas de pré-condição e axiomas de estrutura que afetam o desempenho das tarefas. O problema com estes tipos de linguagens é que seu desenvolvimento é pequeno, para aplicações pequenas e médias complexidade. Para aplicações de maior complexidade deve-se utilizar linguagens de alto nível como C e Java. Existem plataformas como JADE que definem classes e métodos. As classes representam os agentes, e métodos representam, em muitos casos, implementações das questões pertinentes aos sistemas multiagentes (GLUTZ; VICCARI, 2003).

### 2.5.8 Tipos de Ambientes

As interações são eventos que fazem com que os agentes mantenham algum tipo de contato, seja esse contato direto ou indireto, através de outro agente ou do ambiente (Figura 25).

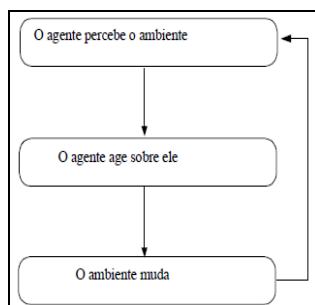


Figura 25. Visão das interações de um agente sobre o ambiente

Algumas propriedades dos ambientes podem variar conforme o modo que o ambiente foi definido e o objetivo da solução a implementar (WOOD, 1994).

- Acessível ou inacessível

O ambiente é acessível se os sensores do agente dão a ele acesso ao estado completo do ambiente. É efetivamente acessível se os sensores detectam todos os aspectos relevantes à escolha da ação. A utilidade de um ambiente acessível é que o agente não precisa manter histórico do ambiente.

- Determinístico ou não-determinístico

Se o próximo estado do ambiente é determinado totalmente pelo estado atual e pelas ações dos agentes, o ambiente é determinístico. Se o ambiente é inacessível, pode parecer não determinístico. Por isso, define-se determinismo do ponto de vista do agente.

- Episódico ou não-episódico

A experiência do agente é dividida em episódios. Um episódio é igual à percepção mais a ação. A qualidade das ações depende somente do episódio. Episódios subsequentes não dependem do anterior. O agente não precisa antever o futuro.

- Estático ou dinâmico

O ambiente é dinâmico se mudar suas propriedades com o passar do tempo. No estático o agente não precisa monitorar o ambiente enquanto decide sobre a ação, também não precisa se preocupar com o tempo.

- Discreto ou contínuo

Se houver um número limitado de percepções e ações distintas definidas, o ambiente é discreto.

#### 2.5.9 Agentes Inteligentes Aplicados no Reconhecimento de Feições

Os sistemas de reconhecimento de feições oferecem uma interpretação de objetos do cenário real que compõe uma imagem, e este é o grande desafio de todo o processo de digitalização, interpretação e classificação de uma imagem. O fato é que, diferentes técnicas para reconhecimento de feições são comumente utilizadas, perante o domínio de diferentes parâmetros por parte dos usuários, a fim de obter-se a representação de uma feição de um tipo de cobertura superficial.

Nos últimos anos tem-se observado uma demanda crescente em todo setor público e privado em matéria de processamento digital de imagem, realidade virtual, agentes inteligentes, etc. Além disso, o desenvolvimento de sistemas e produtos baseados nessas tecnologias muitas vezes requer altas doses de conhecimento e pesquisa.

A complexidade reside na dificuldade que apresenta a análise de uma imagem, que é traduzida em milhões de *pixels* com diferentes intensidades de cor para serem analisados por um sistema. Para realizar esta análise deve-se separar os objetos do fundo da imagem e diferenciá-los entre si. Assim, após o processo de detecção de bordas da imagem, os agentes podem identificar as formas, reconhecer os objetos e alocá-los nas classes correspondentes (ORTEGA, 2008).

Este trabalho propõe a manipulação das informações coletadas pelos agentes especialistas para aplicação no processo de detecção de bordas e consequentemente o reconhecimento de feições de uma imagem. Entretanto é necessário realizar vários testes para

que o agente compreenda e entenda com a rotina do sistema, e o que se quer alcançar, ou seja, a completa manipulação da interpretação e conhecimento sobre a cena da imagem correlacionada com suas informações espectrais. A Figura 26 mostra as funções que desempenham um agente para a manipulação adequada no processo de reconhecimento de feições de uma imagem.

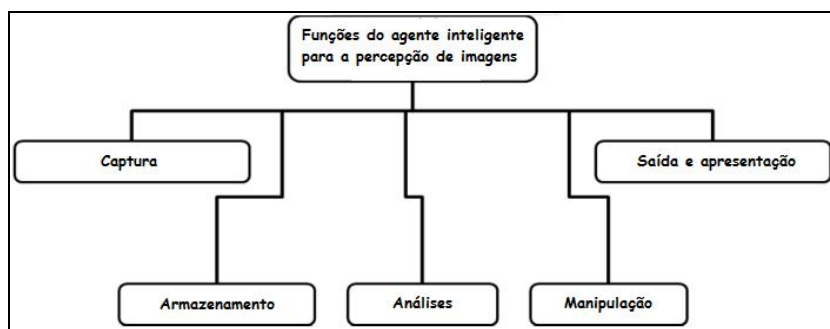


Figura 26. Fluxo de trabalho de agentes inteligentes aplicados ao reconhecimento de feições (ORTEGA, 2008).

Com base na aplicação de agentes inteligentes no campo do processamento digital de imagens as informações em termos de cor, textura, tamanho, tempo e dados socioeconômicos podem ser utilizadas no reconhecimento de feições. Esse processo é constituído pelas seguintes etapas: segmentação da imagem, extração das características, representação, e classificação. Cada etapa cumpre com uma função específica dentro do processo. De todas as etapas, a segmentação é a que permite agrupar regiões de *pixel* homogeneamente. Após a segmentação realizam-se as outras etapas, a extração das características e classificação. Não é possível classificar adequadamente um objeto segmentado incorretamente. A extração das características ocorre sobre as regiões segmentadas, confiando que as regiões geradas durante a segmentação são as únicas que podem conter os objetos de interesse (ORTEGA, 2008).

A problemática do processo de segmentação se acentua quando se pretende reconhecer objetos cujos limites são difíceis de detectar em uma imagem, como é o caso da localização e identificação automática de edificações em área de encosta, constituída predominantemente por sombras, desmatamento ou densa vegetação.

## 2.5 NetLogo

Uma série de linguagens e ambientes está disponível como opções para utilização no desenvolvimento de sistemas inteligentes. A melhor opção é aquela em que todos os critérios para o desenvolvimento do sistema sejam totalmente satisfeitos para o ambiente de interação. Para o trabalho apresentado, foi utilizado a plataforma NetLogo. O NetLogo é um ambiente de modelagem orientada para a simulação de sistemas multiagentes que envolvem um grande número de agentes. A plataforma tem como objetivo fornecer uma multiplataforma de Ambiente de modelagem programável multiagente (PONTAROLO, 2010).

As principais entidades do NetLogo são os ambientes (*patches*), as tartarugas (*turtles*) e o observador (*observer*). O observador controla o experimento, no qual as tartarugas e o ambiente participam ativamente. O ambiente é o agente estacionário, isto é, componente de uma grade em que as tartarugas (que são os agentes móveis) movem-se e interagem.

Os *patches* e as tartarugas podem inspecionar o ambiente ao seu redor, por exemplo, detectar a existência de outros agentes, visualizar o estado de sua vizinhança de *patches* e tartarugas, e modificar o ambiente.

Os agentes podem ser organizados em grupos no mundo da simulação. Provavelmente a característica que mais aumenta a expressividade da modelagem da plataforma é o fato de que cada ambiente e tartaruga podem ter suas próprias variáveis definidas pelo usuário: no caso do ambiente, isto permite modelar ambientes complexos, incluindo um número adequado de variáveis que o descrevem suficientemente. No caso das tartarugas, isto significa simplesmente que cada agente pode transportar seu próprio estado, armazenados novamente em uma série de variáveis definidas pelo usuário.

A linguagem de programação NetLogo permite a especificação do comportamento de cada *patch* e tartaruga, e do controle da execução dos mesmos. A linguagem é simples, expressiva e bastante funcional. Existe um conjunto extenso de programação primitiva, um bom suporte para a matemática de ponto flutuante, números aleatórios e capacidades de plotagem. A programação primitiva inclui, por exemplo, os comandos para mover as tartarugas na grade, os comandos para a inspeção do ambiente (ou seja, o estado de outras tartarugas e *patches*), construção de programação clássica (ramificações, condicionais, repetições), etc. A principal estrutura de dados são as listas (seguindo a abordagem funcional Lisp), e a linguagem suporta tanto as funções de relatórios (*reporters*) como os procedimentos (*procedures*). O monitoramento e a execução dos agentes é controlada pela entidade

observador que pede (*ask*) a cada agente para realizar uma tarefa computacional específica. O ambiente de programação também oferece ferramentas para criação de Interface Gráfica do Usuário (GUI, em inglês *Graphical User Interface*), através da qual as visualizações personalizadas dos sistemas multiagentes podem ser criadas com facilidade (PONTAROLO, 2010).

O NetLogo é uma opção adequada para modelagem de sistemas multiagentes, uma vez que cada agente NetLogo percebe seu ambiente e age sobre ele, carrega seu próprio segmento de controle e é autônomo. O NetLogo é uma boa ferramenta para prototipagem rápida e teste inicial de sistemas multiagentes, bem como uma ferramenta de animação do sistema modelado.

### 3 UM PROTÓTIPO PARA O RECONHECIMENTO DA FEIÇÃO EDIFICAÇÃO

Este trabalho propõe uma nova técnica baseada em um sistema multiagentes para a automática classificação de feições em imagens, com a avaliação através da interpretação da atualização do Cadastro Técnico Imobiliário do município de Angra dos Reis.

A idéia consiste em programar vários agentes com um comportamento específico para o reconhecimento de feições. Em outras palavras, agentes especializados em reconhecer vegetação, agentes especializados em reconhecer vias públicas, como também agentes especializados em reconhecer edificações. Esse último trata-se do principal objetivo da dissertação.

Cada agente, dotado do seu comportamento, irá inferir a classe de feição sobre a qual ele se encontra na imagem. Os agentes usam como ambiente a imagem aérea da região sob estudo, sendo assim, eles fazem uma varredura *pixel a pixel* sobre a imagem aplicando uma análise de acordo com o seu conhecimento programado.

Para a programação dos agentes, foi utilizada a ferramenta NetLogo 4.1.3. Tal ferramenta possui uma área de ambiente 2D sobre a qual os agentes atuam. É possível associar a essa área uma imagem digital, de tal forma que os agentes programados dentro do NetLogo tenham acesso aos elementos dessa imagem, e assim são capazes de interagirem com ela e tirar conclusões conforme a inferência programada na sua base de conhecimento.

Neste capítulo, é apresentado o protótipo do Sistema de Reconhecimento de Feições de Imagem Digital desenvolvido para o reconhecimento de edificações. Na primeira seção, é apresentada uma visão geral dos módulos que compõem o sistema, com suas principais características e entradas.

#### 3.1 Componentes do Sistema de Classificação de Imagem

O sistema de reconhecimento de feições em imagem criado é composto por sete agentes: três agentes de textura para detectar as feições vias, vegetação e edificação; um agente que faz a segmentação; dois agentes que analisam a geometria (proporção e área) do padrão de edificação para confirmar a detecção dos agentes de textura e um agente de conflito



para solucionar os conflitos referentes a classificação do *pixel*. A Figura 27 a seguir mostra a arquitetura do Sistema de Classificação de Imagem.

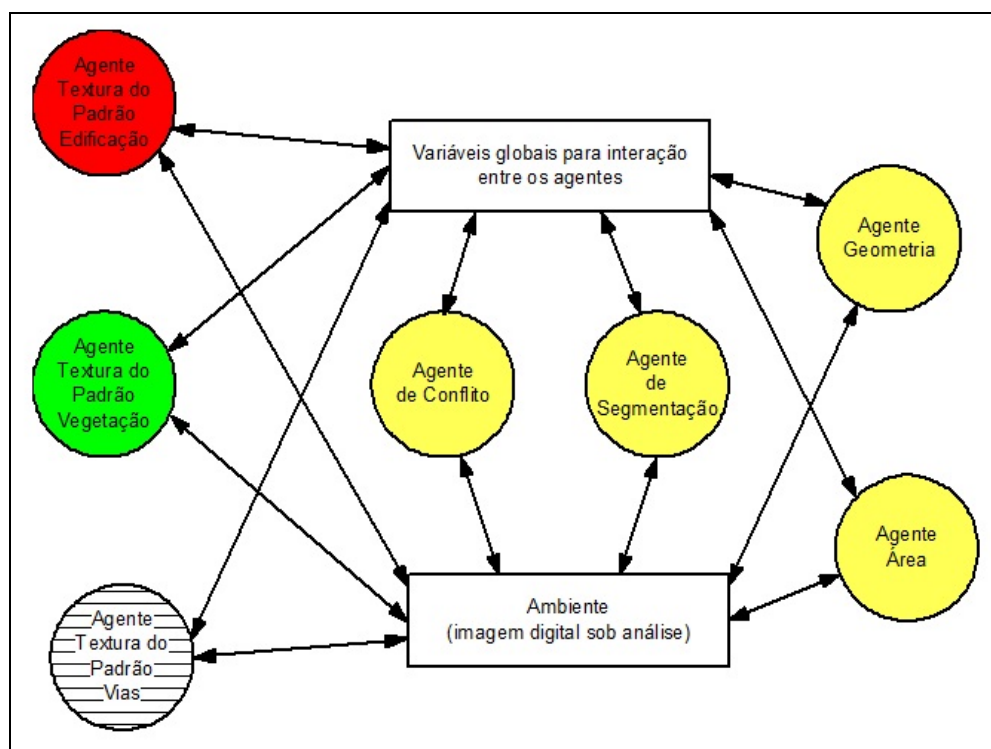


Figura 27. Arquitetura do Sistema de Classificação de Imagem.

O objetivo principal do sistema é o monitoramento da área urbana em expansão, incluindo a área florestal, bem como prover medidas imediatas por conta das instituições competentes, no caso da detecção de uma edificação em área irregular, na tentativa do controle do crescimento urbano desordenado. O modelo do mundo real é representado pelo recorte da ortofoto digital do cenário.

Antes de começar a trabalhar o código na plataforma NetLogo, foi necessário uma base de dados de ortofotos digitais tomadas de duas regiões do município de Angra dos Reis, que caracterizam dois ambientes distintos. Foram feitos dois recortes dessas imagens para que o sistema possa atuar e ser avaliado em diferentes composições de ambiente.

Os agentes percorrem o ambiente da imagem *pixel a pixel*. Tais agentes são especialistas em diferentes funções e possuem comportamentos distintos que acarretam em uma combinação de ações do tipo segmentar e padronizar, tomadas com grau de probabilidade compatível com o seu comportamento.

### 3.2 Contextualização dos Cenários

A presente seção mostra uma contextualização dos cenários que compõem as imagens sob estudo. A principal meta é esboçar um quadro das condições físicas, sociais e econômicas delineadoras das configurações espaciais dos cenários em questão.

O recorte 1 da ortofoto digital apresenta uma área de teste compreendendo uma região caracterizada por um planejamento urbano com lotes, quadras, e ruas com contornos bem definidos e padronizadas. As edificações são de grande porte, possuindo nas adjacências áreas verdes, ruas, estacionamentos, calçadas, etc. (Figura 28a). As edificações possuem tamanhos de áreas entre 90 a 110m<sup>2</sup>. Esta área é plana, litorânea, próxima de Rodovia, rios, e confrontante com outros municípios. O recorte 2 da ortofoto digital é caracterizado por uma região que apresenta um crescimento urbano desordenado. A maior concentração da ocupação urbana é localizada em área de encosta, que é coberta por vegetação densa, próxima ao centro comercial, e a pontos de interesse (Porto, Petrobras, centro comercial, hospital e pontos turísticos). A Figura 28(b) mostra edificações de porte pequeno e com formas irregulares misturadas com a vegetação adjacente. As edificações possuem tamanhos de áreas entre 30 a 70 m<sup>2</sup>. Esta área apresenta edificações, escadarias, vegetação densa e sombras devido à alta declividade. A complexidade que estes objetos estão espacializados dificulta o processo de reconhecimento das feições existentes na cena.

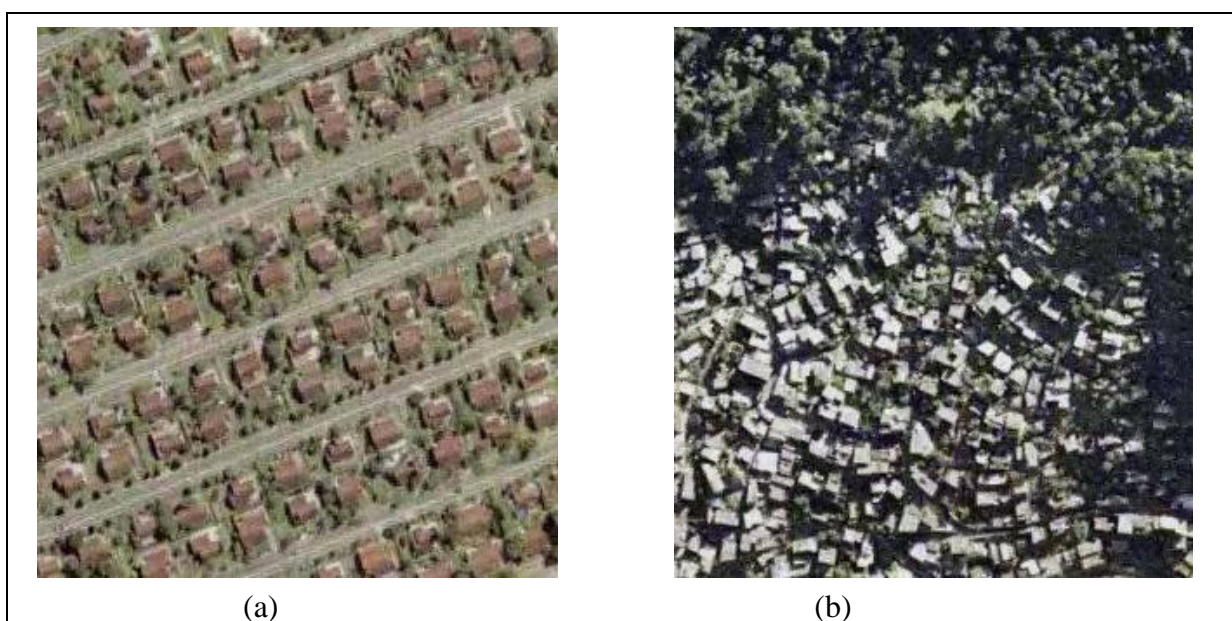


Figura 28. (a) Recorte 1, para área denominada A e (b) Recorte 2, para área denominada B

O recorte 2 apresenta um cenário onde o processo de urbanização do município é dotado de amenidades e situado nas proximidades do centro de produção e consumo, que tem

sido influenciado pela construção de residências secundárias (Figura 29). Tal processo vem condicionando o desenvolvimento local ao capital imobiliário e intensificando os contrastes definidos pela produção de núcleos dotados de externalidades positivas em contraponto à formação de assentamentos desprovidos de infraestrutura, ocupados pelos grupos sociais excluídos.

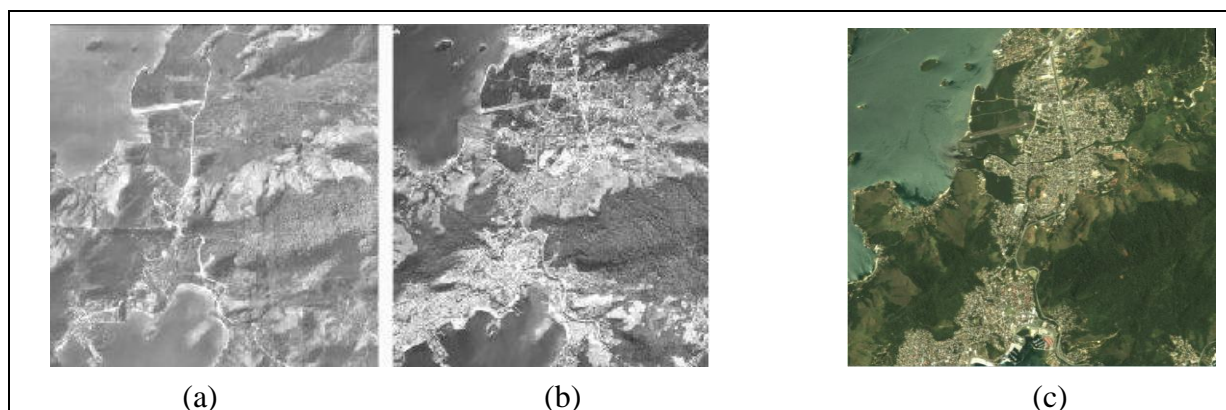


Figura 29. Evolução da ocupação da área central (Centro, ao sul e Japuiba, ao norte) do município de Angra dos Reis: (a) aerofotograma de 1962 (LASA), apresentando ocupação restrita ao Centro; (b) aerofotograma de 1991 (AEROIMAGEM); e (c) Imagem de satélite, IKONOS III de 2007, apresentando uma ocupação urbana em Planície Costeira. (Acervo da Secretaria Municipal de Planejamento de Angra dos Reis).

O crescimento populacional se acelera desproporcionalmente numa curva ascendente a cada década (Gráfico 1). A sociedade local se diversifica em novas camadas sociais. A composição dos trabalhadores agora possui, além dos tradicionais trabalhadores vinculados à pesca e à atividade rural, o operariado vinculado à modernização industrial (metalúrgicos, petroleiros, eletricitários e também portuários, ferroviários), trabalhadores da construção civil, funcionários públicos municipais, comerciários, trabalhadores do setor de turismo e outros serviços. O movimento da ocupação humana é predominantemente a cerca do centro comercial e das áreas de encosta. Essa composição social produz-se no espaço apresentado uma estrutura urbana de maior complexidade e conflitos.

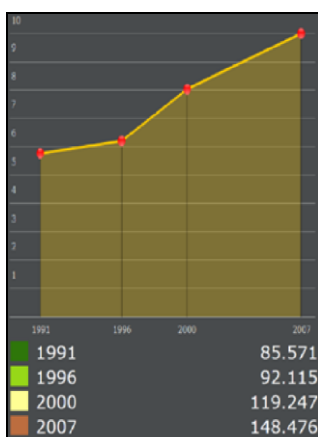


Gráfico 1. Crescimento Populacional de Angra dos Reis, Censos Decenais (IBGE, 2011).

Na Figura 30 é possível visualizar um verdadeiro duelo entre a ocupação humana e os limites da natureza: falta de saneamento, acúmulo de lixo, esgoto, aterros de áreas úmidas, corte de morros, desmatamentos, construções em locais de risco, etc. esse modelo de urbanização ainda está em curso, reproduzindo os mesmos desequilíbrios e eternizando os conflitos sociais e ambientais a ele imputados, e pela crescente massa de imigrantes de baixa renda.



Figura 30. Ocupação humana em área de encostas acerca do centro comercial de Angra dos Reis.

### 3.3 Aspectos do Sistema Multiagente

O propósito do sistema multiagente é a integração de agentes para resolver um problema de complexa resolução. Da interação entre esses agentes resultam as principais características de um sistema multiagente:

- Um único agente não pode solucionar o problema por falta de informação ou capacidade.
- A informação é distribuída, ou seja, ao distribuir e replicar a informação entre os agentes reduz-se a probabilidade de perda de informação.
- A computação é fisicamente distribuída por nós aumentando a capacidade de processamento de informação, sendo assim acrescidas as necessidades de sincronização entre agentes no instante da comunicação.
- Cada agente é auto-organizado para a realização de suas tarefas, em vista que inexistente um sistema centralizado de controle. Os agentes reagem e adaptam-se às alterações do ambiente de forma autônoma.
- Os processos de competição e cooperação são formas de relacionamento entre os agentes e podem conduzir mais facilmente à obtenção do objetivo comum.



Neste trabalho, foi desenvolvido um sistema multiagente, onde cada agente é equipado com uma capacidade de estimar a homogeneidade de uma região em um ambiente. Estes agentes interagem entre si com o objetivo de resolver os conflitos referentes a classificação dos *pixels*. O produto da análise de cada agente é obtido através de uma imagem raster na extensão PGN (*Portable Network Graphics*). Esta imagem é alocada na memória do programa, onde cada *pixel* é um item que compõe uma lista salva na memória do programa. Essa lista comporta uma imagem de tamanho 300x300 *pixels*, ou seja, 9.000 itens. Um número fixo de agentes) é utilizado na análise da imagem. Foram implementados sete agentes (entidades autônomas) diferentes para a classificação da imagem, a saber:

#### Agente Especialista em Textura

Tem a função de identificar e classificar os *pixels* pertencentes a uma faixa de valores RGB prefixada pelo usuário. Previamente a faixa de valor RGB para cada classe de objeto identificado na imagem é fixado de acordo com estudo das cores e interpretação visual do usuário (Figura 31). Este agente analisa a imagem *pixel a pixel*. O *pixel* identificado como pertencente a faixa RGB para uma determinada classe de objeto na imagem é representado na cor vermelha para a classe edificação, na cor verde para a classe vegetação, na cor cinza para a classe vias e na cor marrom para a classe sombra.

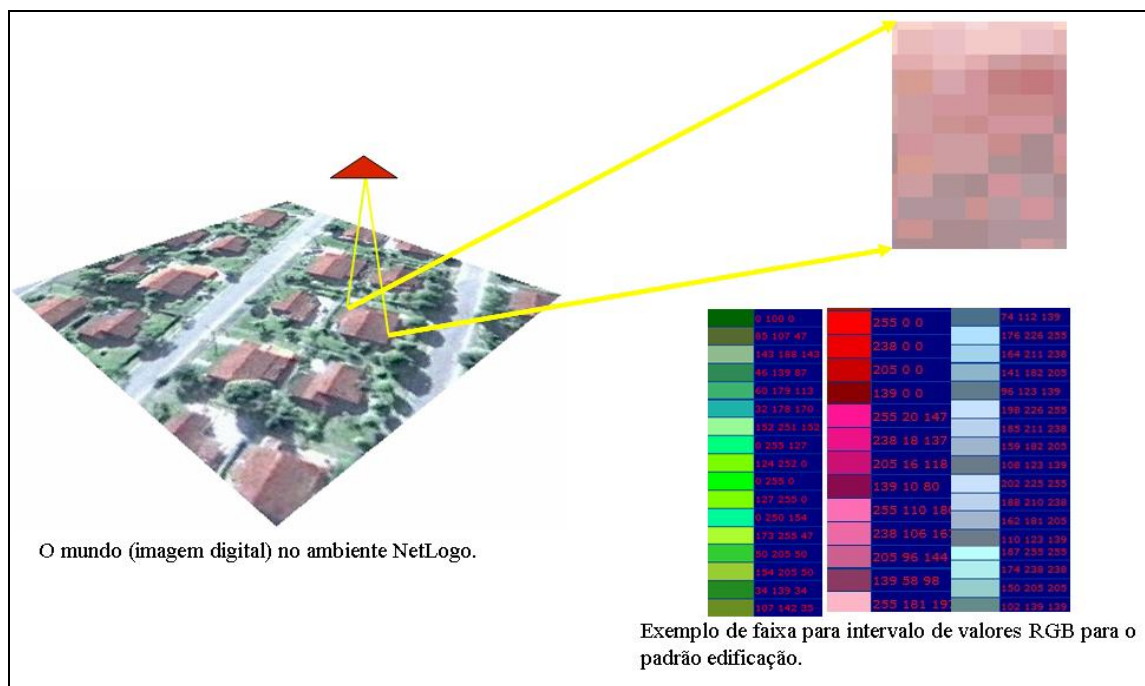


Figura 31. Representação da inferência do agente especialista em textura.

O agente de textura é responsável por realizar a detecção de três padrões de feições: edificação, vegetação e vias públicas. No caso, existem três desses agentes, onde cada um percebe o ambiente (imagem digital) e age sobre o mesmo definindo os níveis de cores da imagem, de forma a identificar a faixa de cor do padrão da feição pelo qual ele é responsável. O agente de textura para edificação identifica a representação RGB do telhado de uma construção existente na imagem. Os outros dois agentes fazem o mesmo para as outras duas feições. A Figura 32 apresenta o diagrama de atividades do comportamento dos três agentes de textura e seu código na linguagem do NetLogo pode ser visto no Apêndice B.

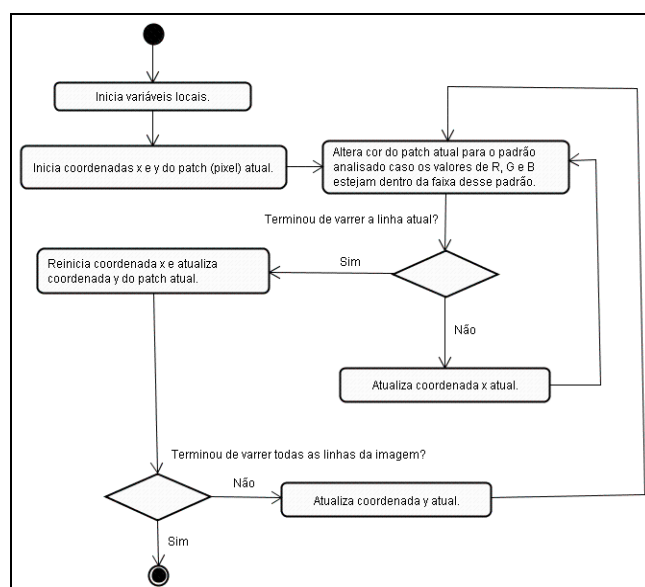


Figura 32. Diagrama de atividades dos agentes especialistas em textura.

### Agente Especialista em Geometria

O agente de geometria confirma, levando em conta parâmetros geométricos, as feições das edificações da imagem aérea detectadas anteriormente pelo agente de textura. Existem dois agentes desse tipo: um deles, chamado de *classifica-edificacao-por-area* (nome do método dentro do NetLogo), confirma a edificação de acordo com a área apenas, enquanto o outro chamado de *classifica-edificacao-por-geometria* (nome do método dentro do NetLogo), confirma a edificação de acordo com a área e com a proporção da feição.

Após o agente textura classificar as áreas pertencentes as faixas de valores das distintas classes de feições, o agente geometria percorre essas áreas homogêneas analisando-as em relação a área (*pixels* que compõem a área vermelha rotulada pelo agente textura) e a proporção (largura e comprimento) da forma dos objetos.

Estes parâmetros são predefinidos pelo usuário, quem obtém o conhecimento da interpretação do cenário analisado, e assim pode-se chegar a tais parâmetros de reconhecimento das feições adaptáveis a diferentes formas e disposições dos objetos nas imagens. Se o padrão da área analisada satisfizer o critério estabelecido pelo usuário, então o agente geometria classifica a área como de padrão que pertence a classe de feição em análise.

Para indicar que uma área já fora reconhecida, o agente geometria colore esta área na cor verde. A cor verde indica que a área na cor vermelha, rotulada, anteriormente pelo agente textura, foi reconhecida pelo padrão da feição em análise, e esta após o reconhecimento realizado pelo agente geometria, foi representada pela cor verde.

A Figura 33 apresenta o diagrama de atividades do comportamento do agente especialista em geometria que avalia o parâmetro de proporção de largura e comprimento do objeto edificação.

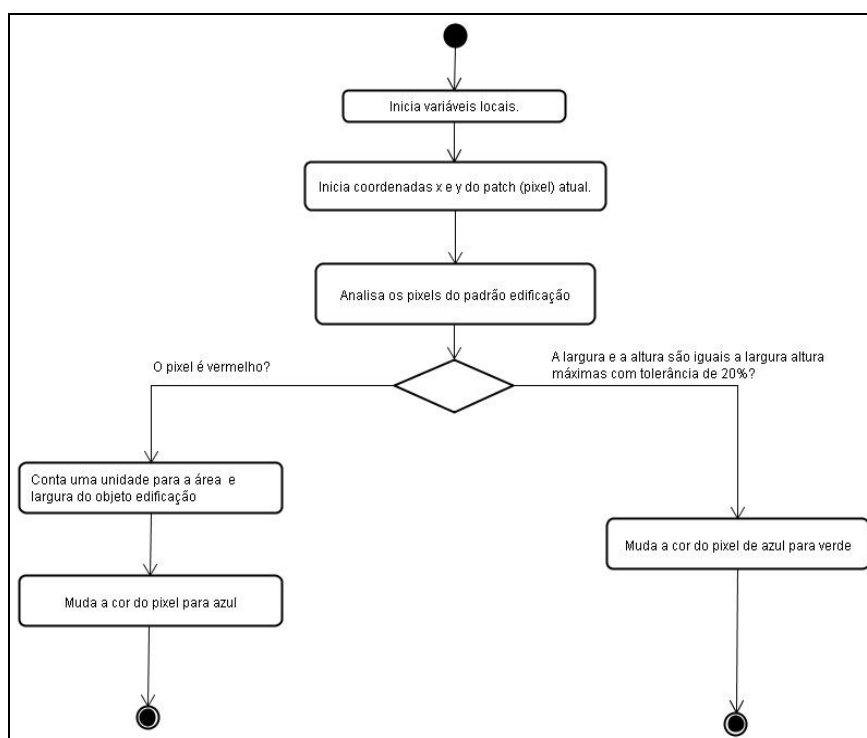


Figura 33. Diagrama de atividades dos agentes especialistas em geometria para proporção do objeto edificação.

A Figura 34 apresenta o diagrama de atividades do comportamento do agente especialista em geometria que avalia o parâmetro área padrão do objeto edificação.

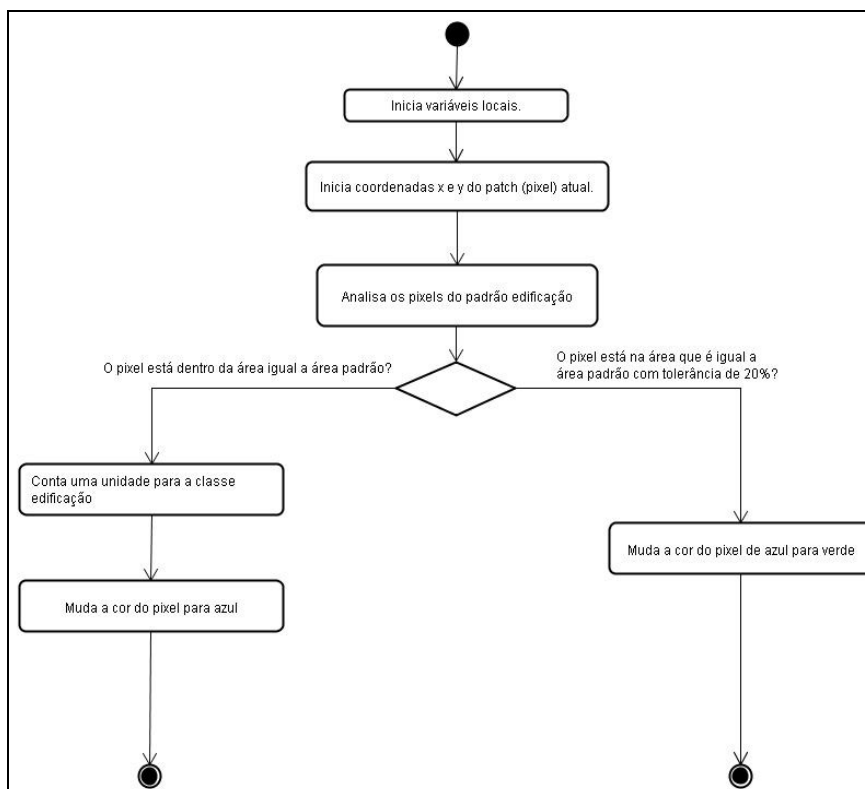


Figura 34. Diagrama de atividades dos agentes especialistas em geometria para área padrão do objeto edificação.

### Agente Especialista em Segmentação

O agente de segmentação tem a função de definir as bordas dos objetos que compõem a imagem através da técnica de limiarização, de forma que os *pixels* homogêneos ou aproximados sejam agrupados de acordo com valor de cinza (Figura 35).

Após carregar a imagem original no mundo (ambiente), o agente segmentação classifica os *pixels* pertencentes ao grupo homogêneo do objeto representados pela cor branca e os *pixels* pertencentes a borda do objeto representados pela cor preta (Figura 33b).





Figura 35. Agente Especialista em Segmentação: (a) Imagem digital onde os *pixels* são representados pelos valores de cores RGB e (b) Imagem digital segmentada, onde as áreas brancas representam os *pixels* próximos homogêneos e as linhas pretas as bordas destas áreas .

A Figura 36 apresenta o diagrama de atividades do comportamento do agente especialista em segmentação que determina as bordas dos objetos da imagem digital.

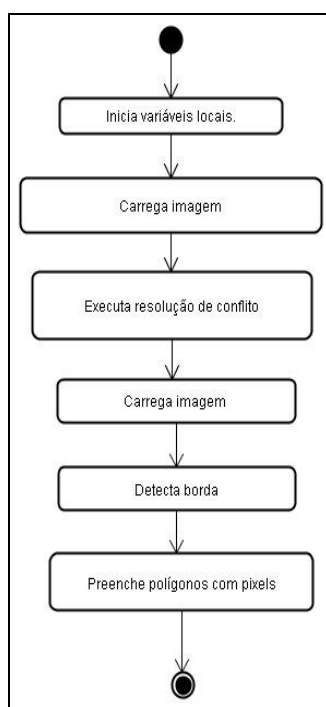


Figura 36. Diagrama de atividades dos agentes especialistas em segmentação.

### Agente Especialista em Conflito

O agente especialista em conflitos executa a função que compara os resultados (imagens) de cada agente. O agente especialista em conflito analisa a imagem *pixel* a *pixel* comparando os *pixel* classificados entre duas imagens, que são resultados do agente textura para duas classes de objeto edificação e vegetação.

Para cada conflito resolvido, o seu resultado é salvo temporariamente na memória do programa em formato de imagem raster na extensão PNG. Esses resultados são recuperados a cada pareamento entre a classe edificação e vegetação para que os conflitos da classificação dos *pixels* sejam resolvidos. O critério para resolução de conflitos de classificação dos *pixels* é a aproximação do *pixel* a borda dos objetos. Esta aproximação é determinada em grau de pertinência do *pixel* a classe analisada, ou seja, quanto mais próximo da borda menor é o grau de pertinência do *pixel* a classe analisada. Após essa resolução de conflitos a imagem armazenada na memória tem representada a classe edificação na cor vermelha, e a classe vegetação na cor verde. Por fim, a imagem original é recuperada e o agente textura é acionado para classificar os *pixels* para a classe vias, que é representada pela cor cinza. Em seguida o agente de conflito compara a imagem salva para o resultado da classificação anterior (classes edificação e vegetação) e compara com a imagem classificada para o padrão de vias. A imagem final classificada, após a execução do agente de conflito, apresenta as classes edificação (cor vermelha), vegetação ( cor verde) e vias (cor cinza).

A Figura 37 apresenta o diagrama de atividades do comportamento do agente especialista em conflitos.

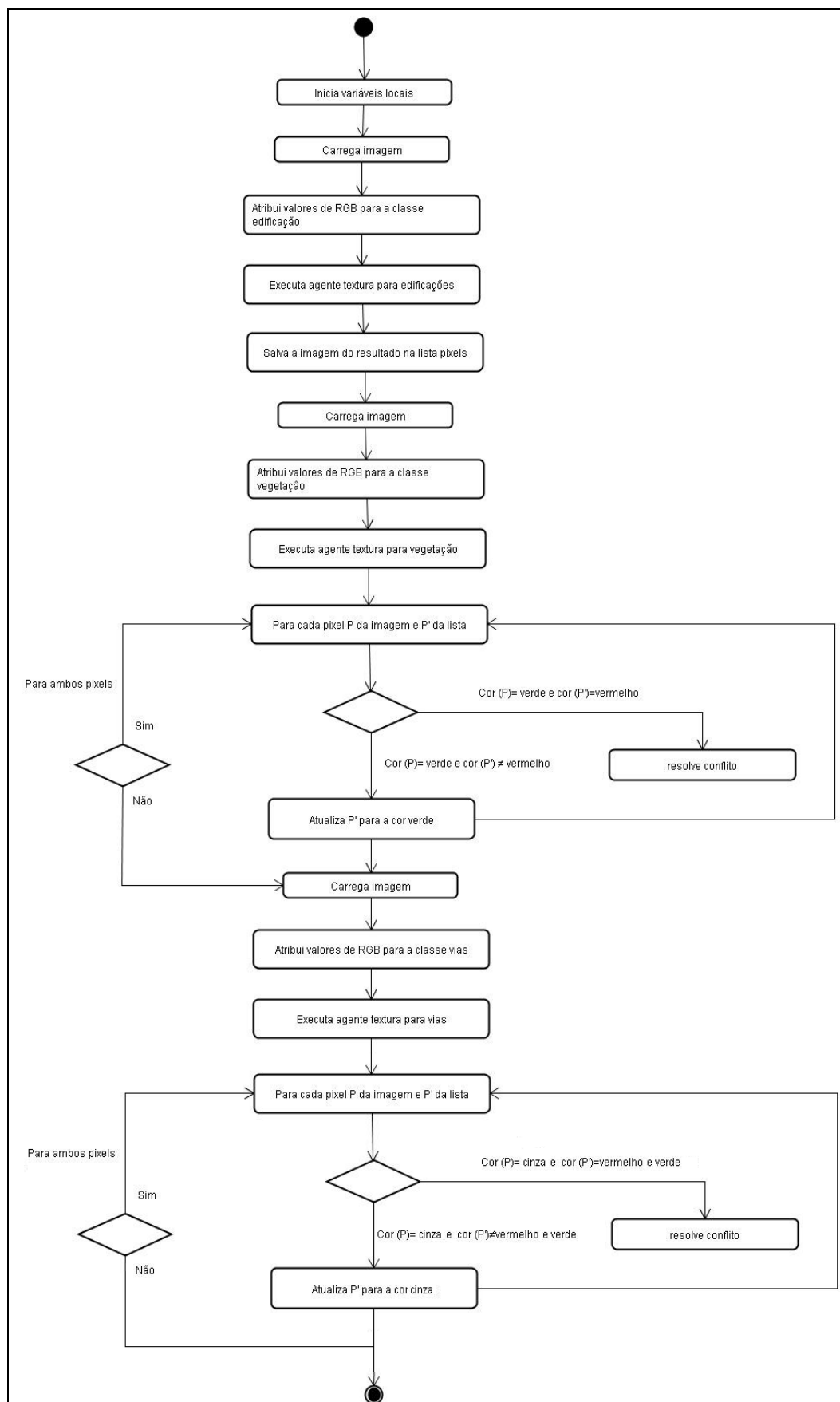


Figura 37. Diagrama de atividades dos agentes especialistas em conflito.

O código na linguagem do NetLogo dos sete agentes implementados no sistema multiagente pode ser visto no apêndice A. O pseudocódigo dos agentes especialistas em geometria e área no Apêndice B.

### 3.4 Interface do Protótipo do Sistema

A fim de facilitar o entendimento do protótipo desenvolvido neste trabalho, a Figura 38 apresenta uma visão da interface do ambiente NetLogo, a qual é composta da seguinte maneira: Uma coluna de sliders para valores R (vermelho), G (verde) e B (azul), um slide para o limiar de segmentação, um botão para carregar a imagem digital original, botões de acionamento dos agentes textura, geometria, segmentação e conflito, e a janela do ambiente (imagem digital) onde os agentes percorrem *pixel a pixel* e apresentam seus resultados.

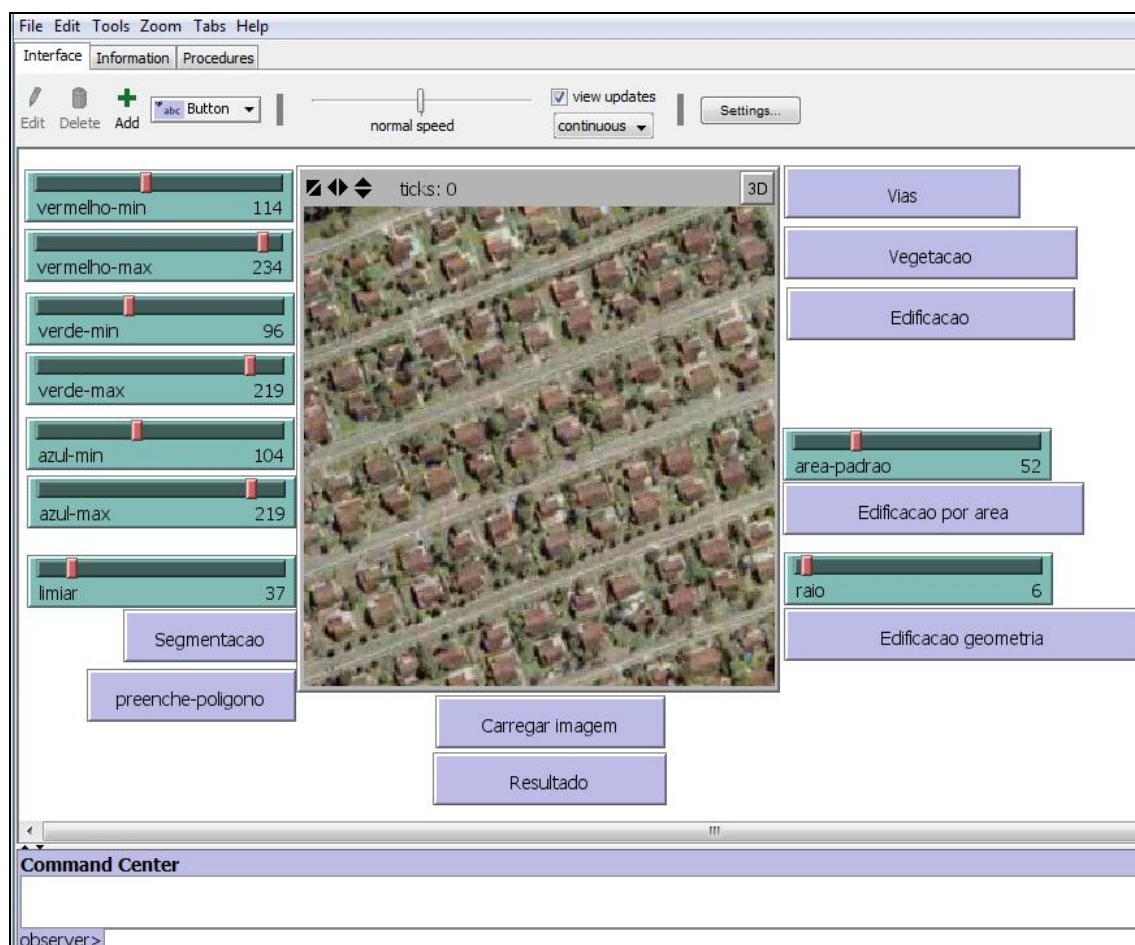


Figura 38. A interface do sistema multiagente no ambiente Netlogo.

A coluna de *sliders* (Figura 39) para valores R,G e B é usada para regular a faixa de valores mínimos e máximos de RGB para o padrão de objeto a ser classificado na imagem. Estes valores para R, G e B são parâmetros para a execução do agente especialista em textura.

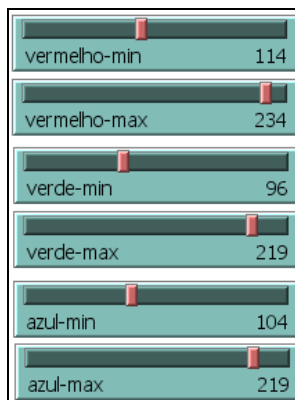


Figura 39. Representação dos sliders para configuração da faixa de valores de RGB.

A janela do ambiente (imagem digital) é o lugar onde os agentes percorrem toda a imagem *pixel a pixel*, ou seja, varrem todas as linhas e colunas da imagem raster (Figura 40). Com a origem do plano de coordenadas da imagem no canto superior esquerdo. Os agentes inferem as linhas no sentido da esquerda para a direita e as colunas de cima para baixo tomando como referencial a origem do plano.

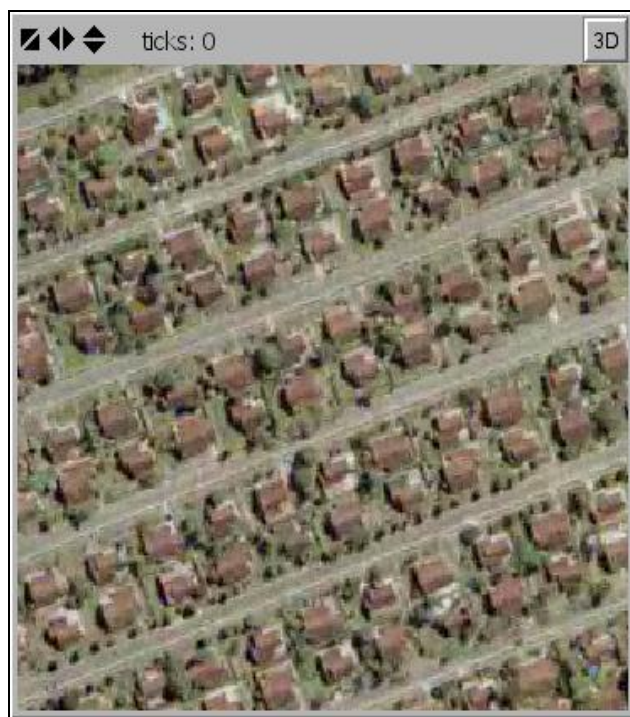


Figura 40. Representação do mundo (imagem digital).

Os botões vias, vegetação e edificação acionam o agente especialista em textura para que este execute suas funções sobre a imagem digital disponível no ambiente (Figura 41). O agente textura percebe a imagem inferindo para cada padrão de objeto a ser classificado. O resultado da imagem classificada para cada padrão de objeto é apresentada no ambiente NetLogo. As imagens dos resultados para cada classe podem ser exportadas na extensão PNG com as classes vegetação, vias e edificações representadas na cor verde, cinza e vermelha respectivamente.



Figura 41. Botões para execução dos agentes especialistas em textura.

O slide área utiliza como parâmetro a área padrão dos objetos para a classe edificação com tolerância também definida pelo usuário. O botão edificação por área aciona o agente geometria que infere a imagem utilizando o parâmetro área padrão para classificar o objeto edificação. O slide raio utiliza como parâmetro a proporção entre a largura e o comprimento máximo do objeto edificação. O botão edificação geometria aciona o agente geometria que infere a imagem utilizando como parâmetro a proporção de largura e comprimento para classificar o objeto edificação (Figura 42).

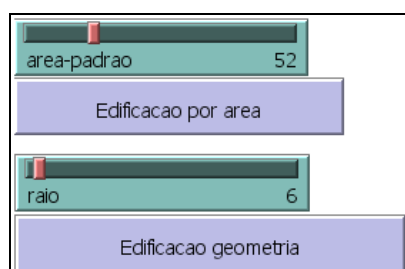


Figura 42. Representação dos sliders que configuram os parâmetros área padrão e geometria utilizados pelo agente geometria para a classificação das edificações.

O slide limiar configura a similaridade dos valores de níveis de cinza da imagem extraindo os objetos de interesse através da seleção de um limiar que separa os grupamentos dos valores de cinza. O botão segmentação executa o agente segmentação (Figura 43).

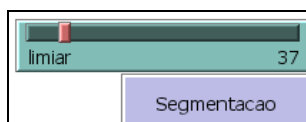


Figura 43. Representação do slide de limiar e do botão para execução do agente especialista em segmentação.

O botão carregar imagem recupera a imagem digital para que esta seja analisada pelos agentes. O botão conflito aciona o agente especialista em conflitos que compara duas imagens a fim de definir a classe a que o *pixel* reconhecido em duas classificações pertence (Figura 44).



Figura 44. Representação dos botões carregar imagem, e conflito que executa o agente especialista em conflito.

O botão salvar armazena a imagem do ambiente no programa NetLogo (Figura 45), através da variável local declarada como lista, onde cada item desta lista é um *pixel* da imagem.



Figura 45. Representação do botão salvar imagem.

## 4 RESULTADOS E ANÁLISES

Este capítulo tem por finalidade apresentar os resultados obtidos com o protótipo desenvolvido para a segmentação e reconhecimento de feições de edificações, bem como os materiais e equipamentos utilizados para o desenvolvimento desse trabalho.

Os resultados obtidos foram a detecção das bordas dos objetos e o reconhecimento das feições vias, vegetação e edificações que compõem os recortes das ortofotos digitais. O reconhecimento da feição edificação é o principal objetivo deste trabalho. A análise foi realizada visual e numericamente, tendo por base comparações entre os resultados obtidos com a utilização do protótipo desenvolvido e os correspondentes resultados obtidos com o uso dos módulos de segmentação e classificação de imagem dos programas ArcGis 10 e o *Spring* 5.1.8. Estes últimos foram obtidos a classificação dos recortes das ortofotos digitais e foram utilizados como dados de referência para a avaliação da técnica proposta. Esses dados encontram-se no mesmo referencial, logo a análise visual foi realizada a partir da comparação entre os grupos de feições (protótipo, ArcGis e *Spring*), simultaneamente, sobre as imagens. Ambos os grupos de feições foram comparados quantitativamente, consistindo em obter as porcentagens da razão de reconhecimento de edificações (REE), proposta por Ruther, Martine e Mtalo (2002) que foi adaptada e utilizada neste trabalho e dada pela seguinte formulação:

$$REE = \frac{EC}{EC + EE} \cdot 100 \quad (19)$$

Onde, EC é o número de edificações reconhecidas corretamente pelo método de classificação e EE é o número de edificações reconhecidas erroneamente pelo método de classificação.

### 4.1 Imagens para Testes

Foram utilizados dois recortes de ortofotos digitais de diferentes regiões do município de Angra dos Reis. Estas ortofotos são do ano de 2006. Estes recortes foram obtidos a partir



do ortofotomosaico digital produzido numa escala de 1:25.000, fornecido pelo Instituto Brasileiro de Geografia e Estatística (IBGE).

A resolução de digitalização de uma imagem é o número de pontos capturados em cada polegada (25,4 mm) do documento. Foram utilizadas para os dois recortes de ortofotos digitais ortofotos com composição nas bandas RGB e 8 bits por *pixel*, ou seja, 256 ( $2^8$ ) níveis de cinza para cada banda *Red*, *Blue* e *Green*. Os recortes das ortofotos digitais no NetLogo possuem tamanhos de área do ambiente para igual a  $(300 \times 300) = 177.590$  pontos por polegada quadrada. Quanto maior a resolução, maiores serão os níveis de detalhes capturados e mais perfeita será a imagem.

Nos testes foram utilizados duas imagens no formato PNG (*Portable Network Graphics*). A grande maioria dos programas de processamento de imagem (que tratam e produzem imagens digitais) abre e salva os formatos PNG e TIFF (*Tagged Image File Format*). O formato TIFF mantém o registro de maneira muito próxima à produzida pelo dispositivo eletrônico que gerou a imagem, seja ele scanner ou câmara digital, devido a isto requer maior espaço de armazenamento e morosidade no tempo de processamento. A plataforma NetLogo não suporta este formato, logo os recortes para testes foram convertidos para o formato PNG.

O formato PNG foi o escolhido na aplicação dos testes realizados por permitir comprimir as imagens sem perda de qualidade, e pela capacidade de definir o nível de cinza de cada *pixel*. Por isso é um formato válido para imagens que precisam manter 100% da qualidade de resolução radiométrica (resolução de níveis de cinza) permitindo assim uma melhor identificação dos objetos.

Testes foram realizados com o formato de imagem JPEG (*Joining Photographic Expert Group*) a fim de otimizar o espaço de armazenamento, e por este formato produzir imagens menores e consequentemente ter rápido processamento para uso em análises sobre um determinado cenário. Porém estas imagens apresentam perda na resolução radiométrica devido ao processo de compressão da imagem. Esta perda produz ruídos nas bordas dos objetos de uma imagem dificultando a utilização deste formato no processo de reconhecimento das feições que compõem uma imagem digital. Embora para um tamanho pequeno de imagem digital os testes não apresentaram uma melhora significativa nos resultados em comparação aos testes realizados com o formato de imagem PNG.

## 4.2 Resultados

Para verificar a adaptabilidade do protótipo desenvolvido foram realizados dois experimentos com áreas distintas para testes conforme mencionado na seção 2.2 (Figura 28).

Na área do recorte 1 a cena envolvida não é de grande complexidade, visto que os objetos na imagem são bem definidos. A forma regular das edificações ajudou para a obtenção de um resultado satisfatório. A Figura 34 mostra os resultados obtidos com a técnica para a detecção de bordas dos objetos da imagem que foi testado para diferentes limiares de segmentação. O agente especialista em segmentação atuou de forma eficiente, extraindo de forma prática e rápida o processo de segmentação da imagem.

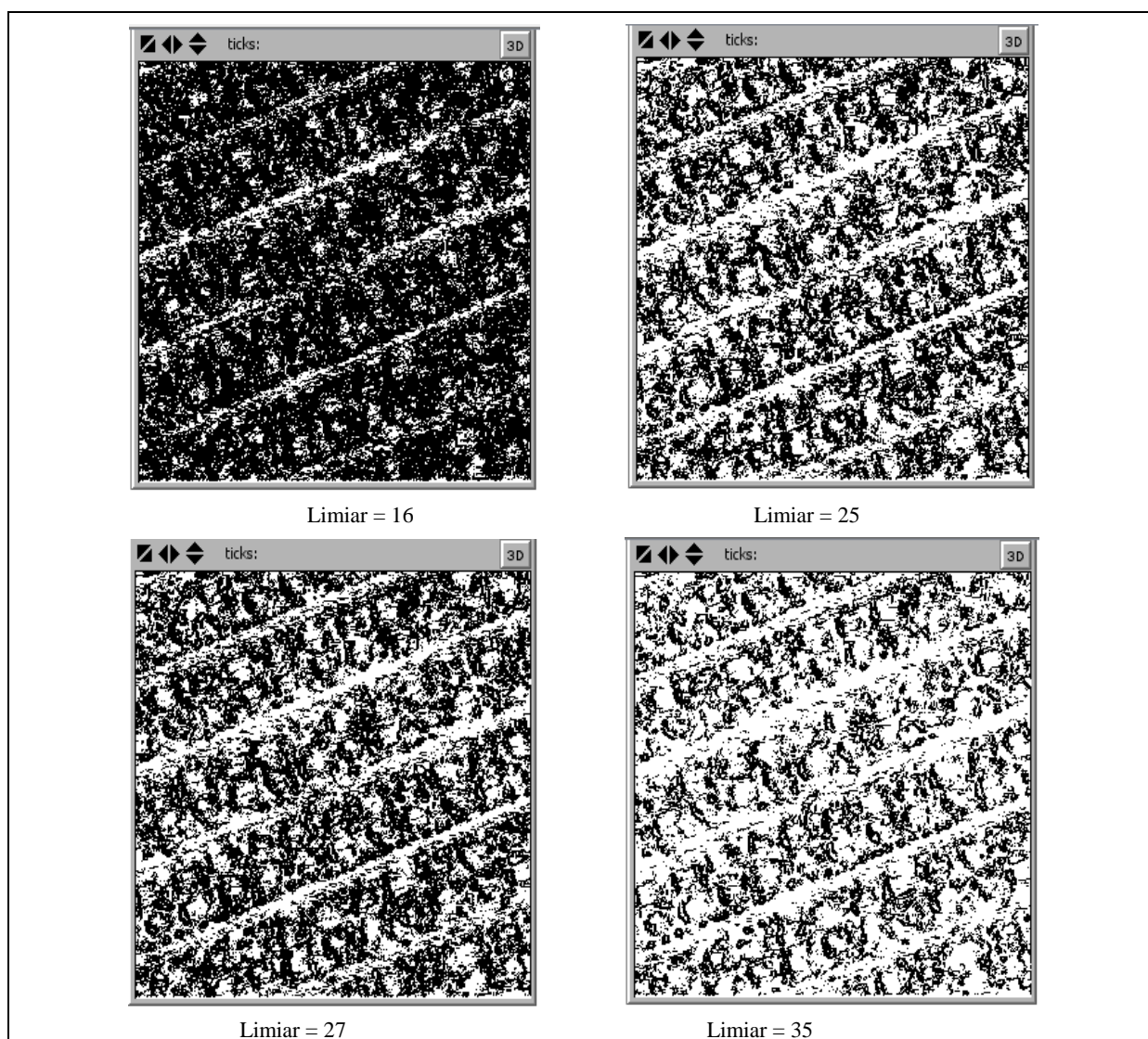


Figura 46. Resultado do agente para segmentação com limiares iguais a 16, 25, 27 e 35 respectivamente.

A Figura 47 mostra o resultado obtido com o reconhecimento das feições vias (Figura 47(a)), vegetação (Figura 47(b)), edificações por cor (Figura 47(c)) e edificações por área (Figura 47(d)). Pode-se verificar nesta Figura que o agente reconheceu as feições satisfatoriamente. No entanto, verificou-se visualmente a dificuldade que o agente teve para agrupar *pixels* da faixa RGB, pré-estabelecida pelo usuário, para a feição edificação e nota-se que algumas edificações perderam área na classificação, ou seja, *pixels* da feição edificação não foram classificados corretamente. Isto se deve pela variação muito intensa entre os níveis de cinza da imagem, e pela perda de resolução espacial da imagem ao se estabelecer o formato JPG. A Figura 47(d) mostra o resultado para o agente classifica edificação-por-área. A imagem apresenta os *pixels* analisados e não reconhecidos como feição edificação na cor azul, e os *pixels* reconhecidos como pertencentes à feição edificação na cor verde.

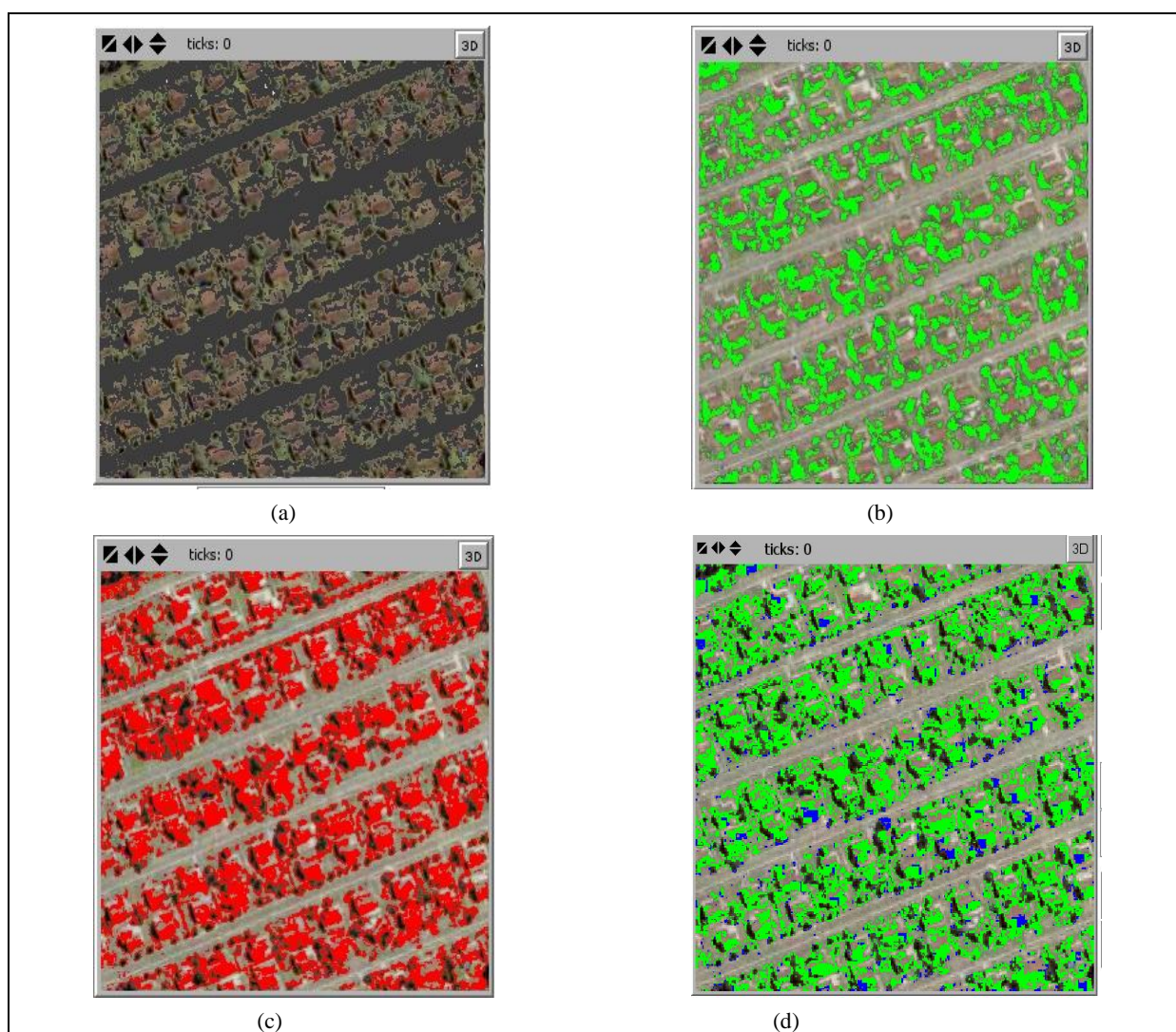


Figura 47. Resultado para o reconhecimento da feição vias, vegetação e edificação da imagem do recorte 1.

No recorte 2, alguns aspectos como a presença de edificação muito próximas umas das outras, prejudicam a extração dos contornos dos objetos. O entorno dificulta o processo de reconhecimento das feições devido ao formato irregular das edificações. Visualmente esta imagem é muito confusa e dificulta até mesmo para o operador humano a identificação correta dos contornos das edificações. A Figura 48 mostra os resultados obtidos com a técnica para a detecção de bordas dos objetos da imagem foi testado para diferentes limiares.

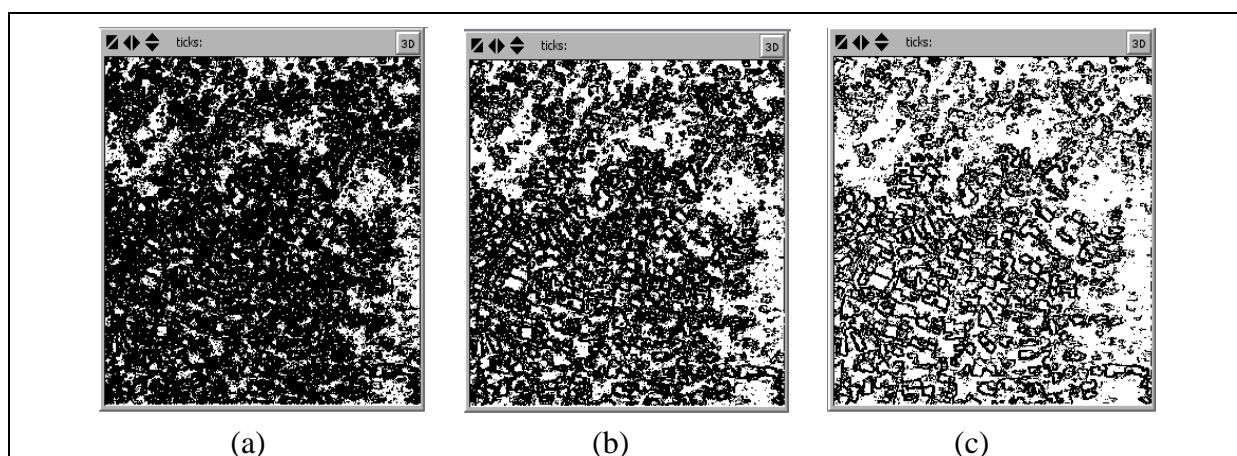


Figura 48. Resultado do agente especialista segmentação: (a) Limiar = 20, (b) Limiar = 30 e (c) Limiar = 45.

A Figura 49 mostra o resultado obtido com o reconhecimento das feições vias (Figura 49(a)), vegetação (Figura 49(b)), edificações por cor (Figura 49(c)) e edificações por área (Figura 49(d)). Pode-se verificar que o agente reconheceu as feições satisfatoriamente. No entanto, verifica-se visualmente a dificuldade que o agente teve para agrupar *pixels* da faixa RGB, pré-estabelecida pelo usuário, para a feição edificação e nota-se que algumas edificações perderam área na classificação, ou seja, *pixels* da feição edificação não foram classificados corretamente. Isto se deve pela grande variação de contraste de níveis de cinza da imagem e pela presença de ruídos nas bordas dos objetos.

A Figura 49(d) mostra o resultado para o agente classifica edificação-por-área. A imagem apresenta os *pixels* analisados e não reconhecidos como feição edificação na cor azul, e os *pixels* reconhecidos como pertencentes à feição edificação na cor verde.



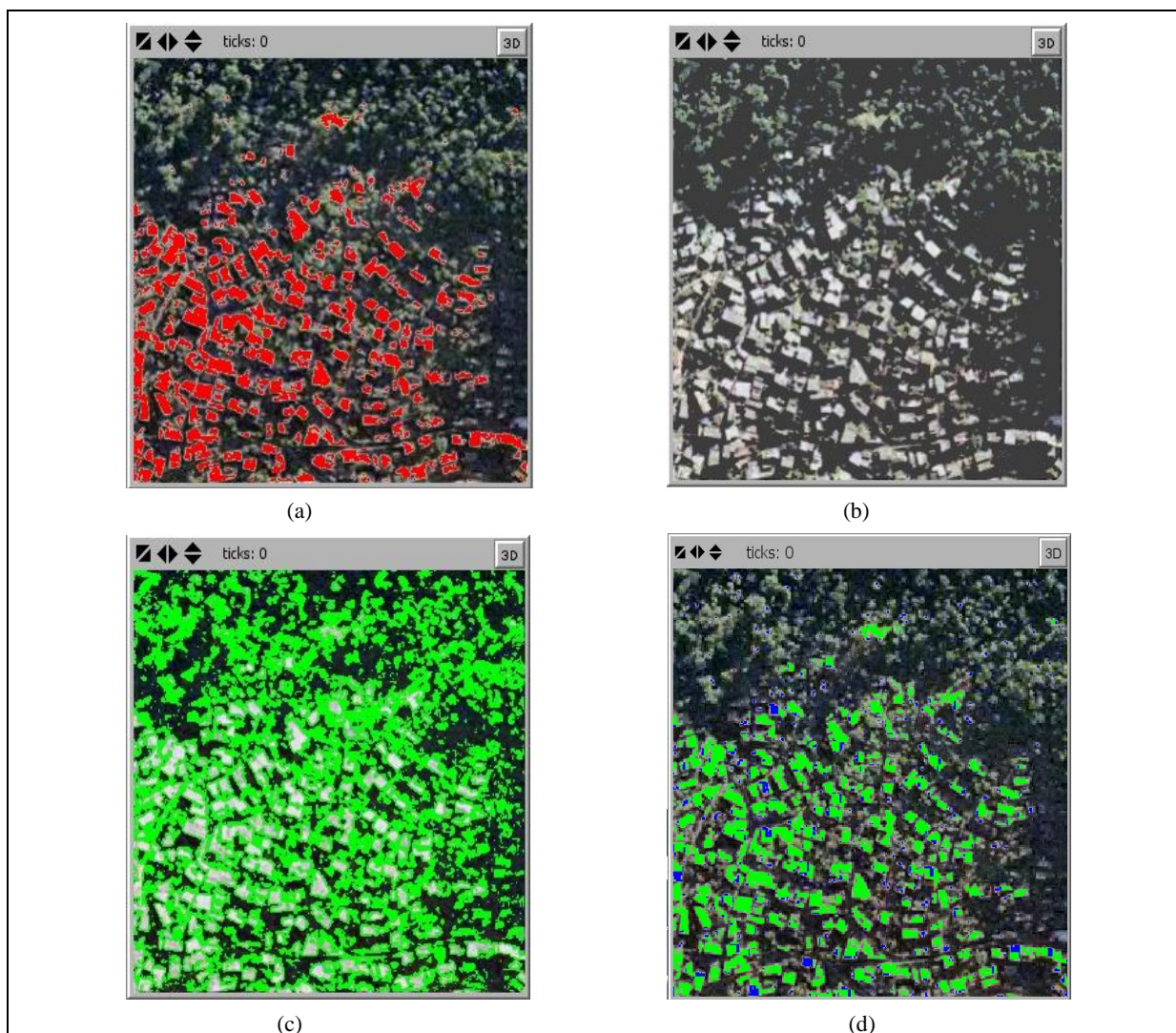


Figura 49. Resultado para o reconhecimento da feição vias, vegetação e edificação da imagem do recorte 2.

A Tabela 3 mostra os resultados numéricos obtidos com o método de reconhecimento da feição edificação pelo protótipo desenvolvido.

Tabela 3 - Análise numéricas dos resultados no protótipo.

	EC	EE	REE(%)
Recorte 1	93	12	88,6
Recorte 2	144	27	84,2

Para uma análise visual mais consistente, a Figura 50 mostra todas as feições reconhecidas nas duas imagens de teste (recorte 1 e 2), através da nova técnica proposta neste trabalho. Após a classificação dos agentes especialistas em textura, geometria e segmentação o agente especialista em conflito executa as suas funções para resolução de conflitos de classificação de *pixel*. Como resultado final obteve-se uma imagem classificada: as classes de

feições edificações estão identificadas pela cor vermelha, as vias na cor cinza e a vegetação na cor verde.

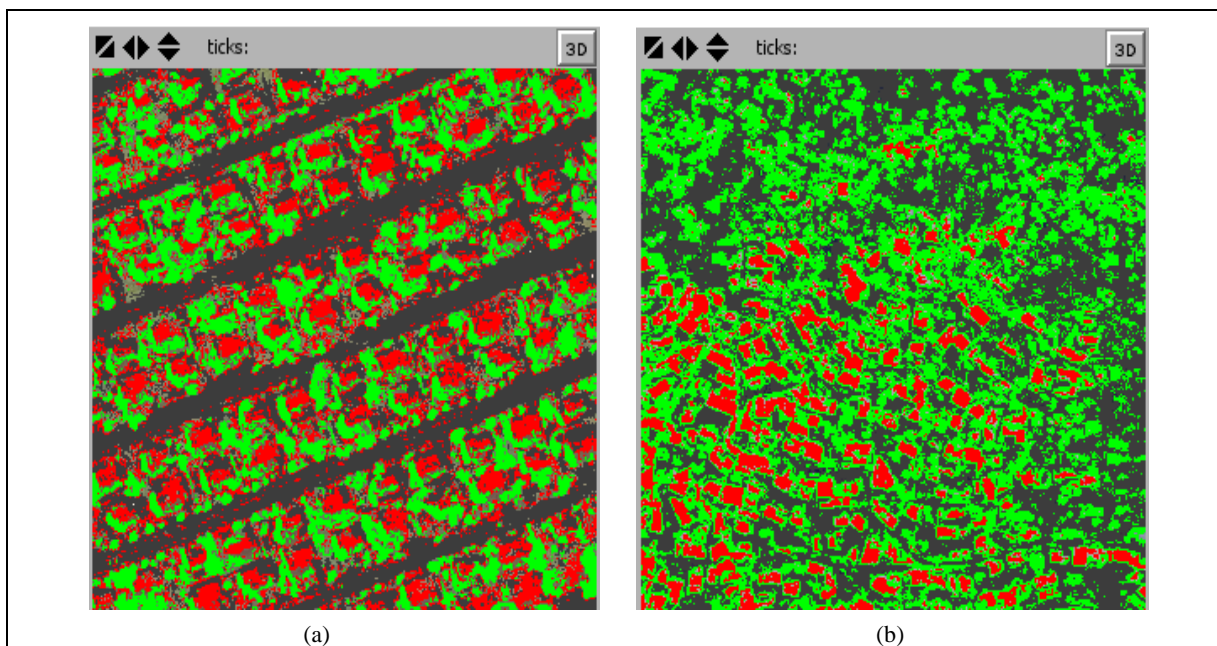


Figura 50. Resultado da classificação final. (a) imagem do recorte 1 e (b) imagem do recorte 2.

#### 4.3 Avaliação do Protótipo

Para avaliar a abordagem proposta com outras abordagens de classificação de imagem, foram feitos outros dois experimentos com as áreas testes em programas convencionais.

Ao realizar uma análise visual nos resultados apresentados pelo método de classificação de imagem pelos programas ArcGis 10 e *Spring* 5.1.8, verifica-se que algumas edificações vizinhas fundiram-se, resultando em uma única feição de edificação. Isso provavelmente se deve ao fato da sombra dos objetos ou pela baixa variação de níveis de cinza entre os vizinhos.

A Figura 51 mostra as imagens dos recortes classificadas pelo programa ArcGis. Na Figura 51(a) foram reconhecidas as feições das edificações (em rosa), da vegetação (em amarelo), e das vias (em laranja) do recorte 1. Na Figura 51(b) mostra as feições edificações (em rosa), vegetação (em verde), e sombra (em cinza) reconhecidas no recorte 2. A Figura 51(b) mostra edificações de porte menor misturado com a vegetação adjacente, no entanto foi possível separar as edificações dos outros objetos adjacentes. Isso mostra que o método utilizado pelo programa ArcGis conseguiu separar eficientemente as edificações, apesar de

não ter preservado sua forma real. Os resultados obtidos neste experimento são consistente no que se refere à extração das feições nos recortes de teste.

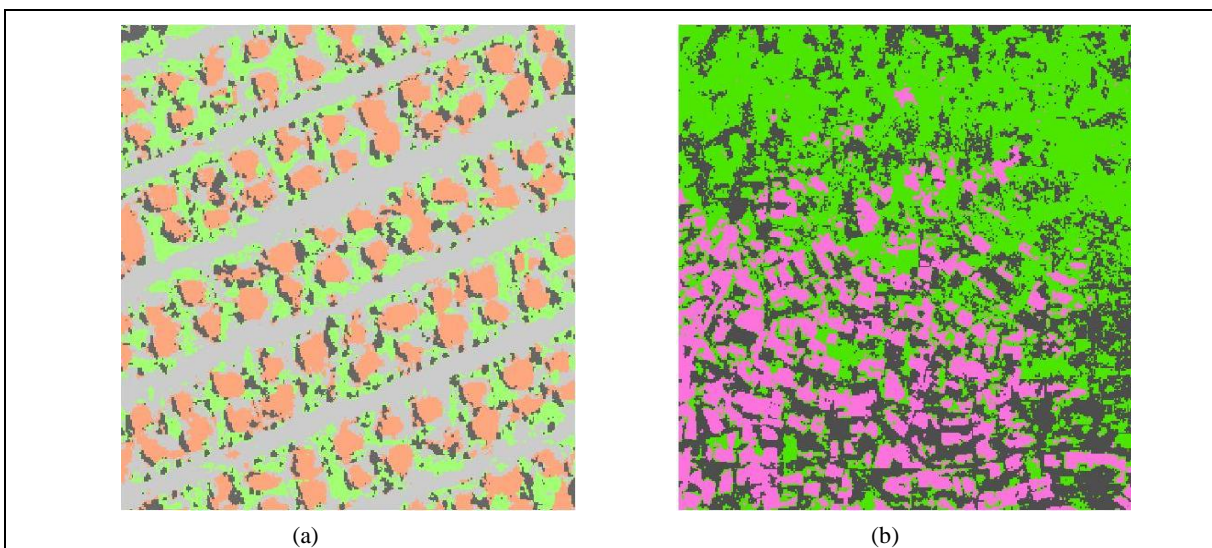


Figura 51. Resultados da classificação de imagem por máxima verossimilhança realizado pelo programa ArcGis 10: (a) Recorte 1 e (b) Recorte 2.

A Tabela 4 mostra os resultados numéricos obtidos com o método de reconhecimento da feição edificação pelo programa ArcGis.

Tabela 4 - Análise numéricas dos resultados no ArcGis.

	EC	EE	REE(%)
Recorte 1	101	4	96,2
Recorte 2	164	7	95,9

A Figura 52 mostra as imagens classificadas pelo programa *Spring* para as feições edificações (em azul), vegetação (em verde), e vias (em cinza) do recorte 1.



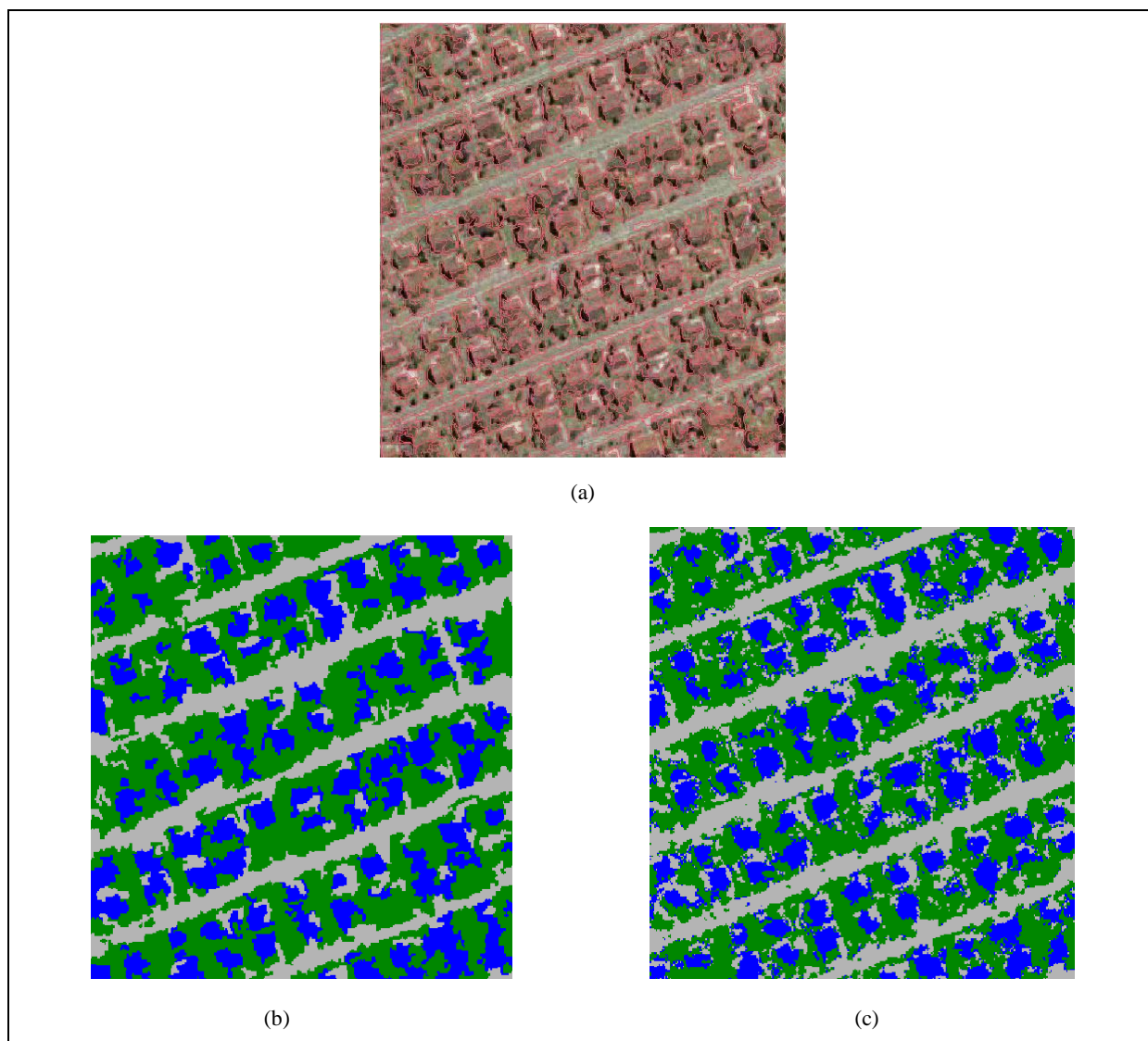


Figura 52. Resultados da segmentação e classificação de imagem realizada pelo programa *Spring* 5.1.8: (a) Segmentação com limiar de área = 10 e limiar de similaridade = 25 (b) Classificação por regiões através do algoritmo Bhattacharya e (c) Classificação por *pixel* através do algoritmo máxima verossimilhança.

A Figura 53 mostra as imagens classificadas pelo programa *Spring* para as feições edificações (em azul), vegetação (em verde), e sombra (em cinza) do recorte 2.



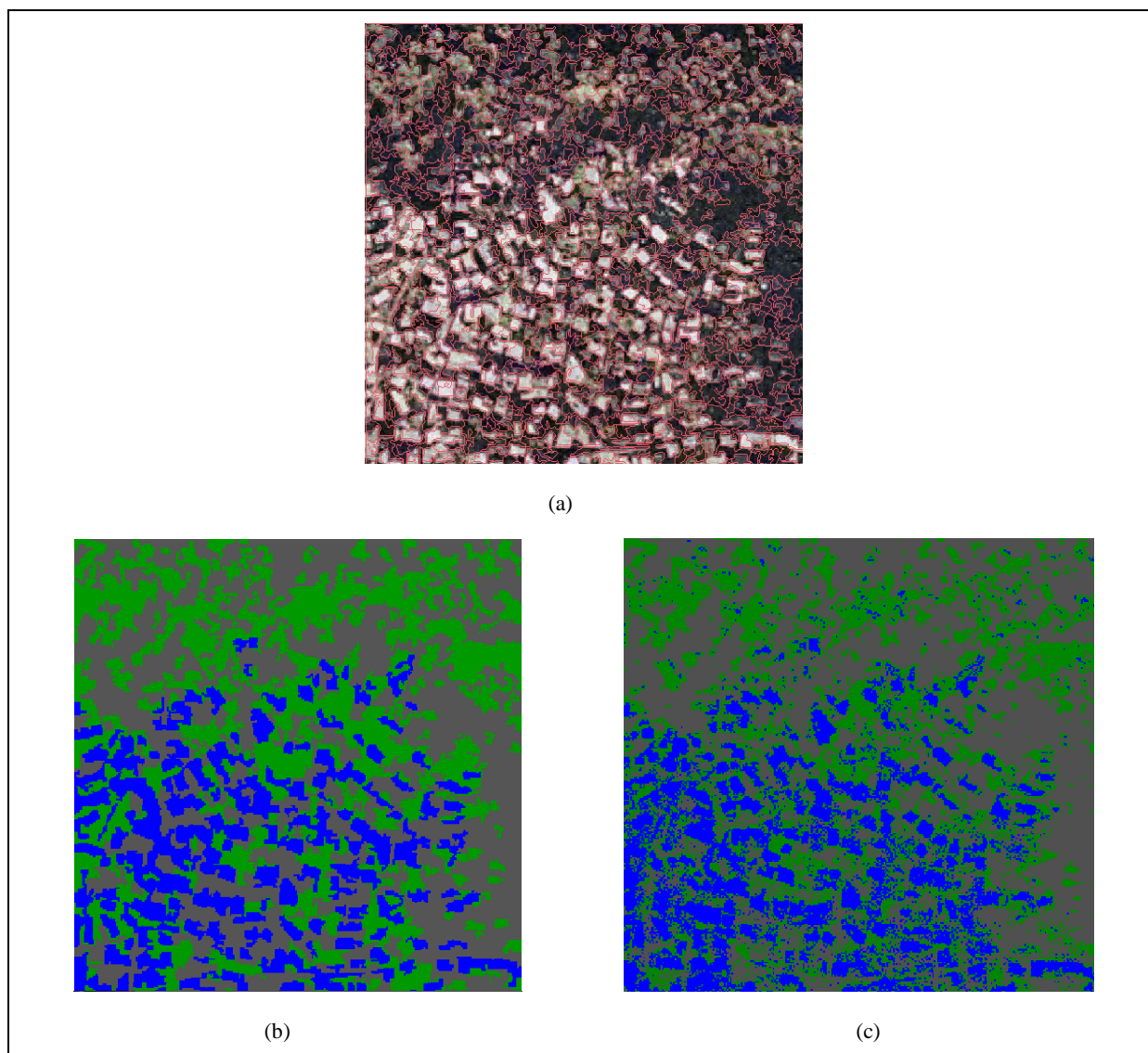


Figura 53. Resultados da segmentação e classificação de imagem realizada pelo programa *Spring* 5.1.8: (a) Segmentação com limiar de área = 10 e limiar de similaridade = 25 (b) Classificação por regiões através do algoritmo Bhattacharya e (c) Classificação por *pixel* através do algoritmo máxima verossimilhança.

A Tabela 5 mostra os resultados numéricos obtidos com o método de extração da feição edificação no programa *Spring*.

Tabela 5 - Análise numéricas dos resultados no *Spring*.

	EC (%)	EE(%)	REE(%)
Recorte 1	103	2	98,1
Recorte 2	165	6	96,5

Para comparar quantitativamente os resultados, considerou-se a quantidade de edificações reconhecidas por cada técnica utilizada (REE), e o deslocamento das bordas dos

objetos verificado em todos os experimentos. A Tabela 6 mostra os resultados numéricos obtidos com o método de extração da feição edificação pelos três programas avaliados.

Tabela 6 - Resultados dos testes nos programas NetLogo (Protótipo), ArcGis e Spring.

Imagem \ Programa	Protótipo (%)	ArcGis (%)	Spring (%)
Recorte 1	88,6	96,2	98,1
Recorte 2	84,2	95,9	96,5

O primeiro experimento utilizou o protótipo desenvolvido neste trabalho. Analisando visualmente as feições extraídas, é possível perceber as irregularidades nas bordas das feições. No entanto, os bons resultados fornecem evidência de que essas irregularidades nas bordas das feições reconhecidas, não interferiram significativamente na análise visual dessas feições. Na análise numérica para este experimento, o parâmetro REE de edificações reconhecidas foi de 88,6% para o recorte 1 e 84,2 para o recorte 2, e apresentou o valor de edificações reconhecidas erroneamente de 11,4% para o recorte 1 e para o recorte 2 o valor de 15,8%. Cabe ressaltar que, na área teste do recorte 2, onde as edificações estão agrupadas e suas bordas não estão bem definidas, os parâmetros utilizados pelo agente geometria, área e proporção, não prejudicaram de maneira significativa o reconhecimento da feição edificação, visto que os atributos retangularidade e área credenciaram este tipo de feição como sendo edificação. Este experimento mostrou um bom desempenho do método em áreas com edificações espacialmente bem distribuídas, conforme pode-se verificar no resultado para o reconhecimento da feição edificação no recorte 1.

No segundo experimento, foi utilizado o programa ArcGis para o reconhecimento das edificações. Nesse experimento verificou-se resultados próximos aos valores dos resultados do programa Spring que é considerado o mais eficiente para a classificação de imagens digitais. O parâmetro REE foi de 96,2% para o recorte 1 e 95,9% para o recorte 2.

Observou-se que, no terceiro experimento, com o uso do programa *Spring*, o método de reconhecimento de feições foi eficiente, visto que o parâmetro REE obteve o maior valor, isto é 98,1% para o recorte 1 e 96,5% para o recorte 2.

Conclui-se então de forma geral que a técnica desenvolvida possibilitou o reconhecimento das edificações de forma satisfatória, mas com a limitação principal de que as feições reconhecidas são aproximações para as respectivas edificações.

## 5 CONCLUSÕES

Neste trabalho foi apresentado um protótipo de um sistema para reconhecimento de feições edificações, vias e vegetação, sendo o reconhecimento da feição edificação o principal objetivo deste trabalho. Este sistema é baseado em agentes inteligentes e implementado na plataforma NetLogo.

Este método baseia-se em sete agentes inteligentes. Um agente foi criado para segmentar a imagem, ou seja, detectar as bordas dos objetos da imagem, três agentes foram criados para reconhecer as feições edificação, vias, vegetação, respectivamente, um agente para reconhecer a feição edificação pelos parâmetros geométricos e um agente especialista em conflitos para resolver problemas na classificação dos *pixels* de uma imagem. Foram realizados dois experimentos com dois recortes de diferentes cenários das ortofotos digitais do município de Angra dos Reis, os quais forneceram subsídios para a análise do desempenho da técnica proposta. A escolha das áreas de teste levou em conta a complexidade das configurações dos objetos presentes na cena. Desta forma, foram selecionadas uma área teste com edificações isoladas e outra com agrupamentos de edificações. Essa escolha teve como principal objetivo verificar a robustez do sistema.

A proposta deste trabalho foi apresentar um sistema com interface amigável, de fácil manipulação para identificar os objetos que compõem a cena de uma imagem, de modo a facilitar uma análise mais aprofundada, seja por um usuário comum (humano) ou um sistema de visão artificial. O protótipo foi desenvolvido considerando a finalidade: facilitar a análise e a detecção de edificações a partir de imagens digitais, destinando-se como ferramenta para usuários (analistas, gestores, usuário comum) sem conhecimento prévio das técnicas de processamento digital de imagem. Para isso, foi desenvolvida uma interface através da qual o usuário comum pode utilizar distintas funções para o processamento de segmentação e reconhecimento de feições.

As aplicações de agentes inteligentes são muito diferentes, visto que as necessidades criadas pela demanda enfatizam a prioridade para obter melhores resultados na implementação de ferramentas para coleta de informações, a fim de converter os processos complexos em tarefas que podem ser realizadas por quaisquer usuários. Com a utilização de agentes na implementação do protótipo para obtenção de informações dos objetos que compõem uma imagem, pretendeu-se potencializar os processos do mesmo, criando um sistema agradável e confiável para utilização no processo de tomada de decisão.

Ao realizar uma análise visual dos resultados apresentados pelo método de reconhecimento de feições proposto neste trabalho, verifica-se que algumas edificações vizinhas fundiram-se, resultando em um único contorno de edificação. Isso provavelmente se deve ao fato das sombras dos objetos ou à baixa variação de níveis de cinza entre os vizinhos. Apesar de o recorte 2 conter edificações de porte menor do que as edificações do recorte 1, e misturadas com a vegetação adjacente, foi possível separar as edificações dos outros objetos adjacentes nos dois recortes de testes. Isso mostra que o método utilizado conseguiu separar eficientemente as edificações, apesar de não ter preservado sua forma real. Os resultados obtidos neste experimento são consistentes no que se refere à extração das feições nos recortes de testes.

O sistema apresentado para segmentação e reconhecimento de feições teve um rápido desempenho e mostrou-se simples em sua elaboração e compreensão. A plataforma NetLogo facilitou consideravelmente a programação da nova técnica de segmentação e reconhecimento de feição.

A técnica apresentada exigiu um menor esforço do usuário para obter resultados com os níveis de concordância compatíveis com a realidade. Isto se deve ao fato da não exigência de ser gasto muito tempo em outras etapas como, por exemplo, a etapa de campo para a seleção de amostras, ocasionando, desta forma, uma economia considerável de tempo.

Conclui-se então, de forma geral, que o protótipo desenvolvido possibilitou a extração das bordas dos objetos da imagem de forma satisfatória, mas com a limitação principal de que os polígonos extraídos são aproximações para os respectivos objetos. A contribuição principal deste trabalho está relacionada com a exploração de relações espaciais entre as edificações na cena, o que foi viabilizado pelo reconhecimento das feições da imagem.

Para concluir, pode-se considerar que o protótipo teve um bom desempenho, apesar de não ter obtido resultados próximos ou melhores do que os dois programas (ArcGis e *Spring*) utilizados nos testes para avaliação, cumprindo assim o principal objetivo que é proporcionar ao usuário um programa com uma interface simples, com menos interações e parâmetros técnicos prefixados pelo usuário para o processamento da classificação da imagem. Outros programas demandam etapas importantes e que requerem uma gama de informações e interações por parte do usuário, tornando o programa científico e de difícil manipulação para usuários comuns. O protótipo desenvolvido apresenta bom potencial e adaptabilidade para o uso em diferentes análises e imagens, podendo ser a base para a geração de mapas de uso da terra de uma forma mais rápida do que a classificação convencional.

## 5.1 Trabalhos Futuros

Os trabalhos científicos relacionados a segmentação de imagens estão dirigidos para uma melhor precisão, exatidão e velocidade computacional das técnicas existentes, assim como a redução da quantidade de interação manual. A precisão e exatidão podem ser melhoradas incorporando informações prévias e combinando aos métodos discretos e contínuos. Para incrementar a eficiência computacional, considerou-se o NetLogo uma boa plataforma para implementação para sistemas multiagentes, e, portanto, para o protótipo do sistema de reconhecimento de feições de imagem.

Pode-se concluir que a arquitetura autônoma e adaptável de sistema de reconhecimento de feições é capaz de concorrer ou superar os métodos de segmentação e classificação de imagens atuais, pelos indicadores dos experimentos realizados. A arquitetura demonstra ser promissora e competitiva com as outras encontradas atualmente, e tem o potencial de superar ainda mais seu resultado a partir de otimizações a serem implementadas em trabalhos futuros.

Apesar de o formato TIFF preservar melhor a resolução espacial de um objeto na imagem, devido às limitações do NetLogo, que não aceita imagens nesse formato, os recortes foram convertidos do formato TIFF para PNG, que preserva a qualidade da resolução radiométrica dos *pixels* e é reconhecido pelo programa NetLogo.

O tamanho dos recortes de testes foi, inicialmente, limitado a dimensões pequenas, assim o protótipo foi testado apenas para imagens com nível limitado de informações. Portanto, seria interessante a adequação deste sistema para testes em imagens no formato TIFF e com escalas maiores de informações. Além disso, é possível concluir que a adição de agente especialista em análise multitemporal (imagens de diferentes épocas) e agente especialista em socioeconômica (informações socioeconômicas da região em análise), entre outros agentes com funções restritas ou abrangentes, possibilitará o sistema a atingir o objetivo de realizar a classificação de uma imagem digital.

## REFERÊNCIAS

- ANGELOTTI, E. S. Utilização da lógica paraconsistente na implementação de um sistema multiagente. 2001. 75 f. Dissertação (Mestrado em Informática aplicada) - Programa de pós-graduação em informática aplicada, Pontifícia Universidade Católica do Paraná, Curitiba, 2001.
- ANTUNES, A. F. B. Classificação de Ambiente Ciliar Baseada em Orientação a Objeto em Imagens de Alta Resolução Espacial. Tese (Doutorado em Ciências Geodésicas). Setor de Ciências da Terra, Universidade Federal do Paraná, Curitiba, 2003.
- BAATZ, M.; SCHAPE, A. *Multiresolution segmentation: An optimization approach for high quality multi-scale image segmentation*. In: *Angewandte Geographische Informationsverarbeitung, Wichmann - Verlag*. Proceedings. Wichmann-Verlag: Heidelberg, 2000.
- BEAUCHEMIM, M.; THOMSOM, K. P. B. *The Evaluation of Segmentation Results and the Overlapping Area Matrix*. *International Journal of Remote Sensing*, v. 18, nº. 18, 1997.
- BRICE, C. R.; FENEMA, C. L. *Scene Analysis Using Regions*. Artificial Intelligence. v. 1, 1970.
- CASTLEMAN, K. R. *Digital Image Processing*. Prentice-Hall, USA, 1996.
- CHUVIECO, E. *Fundamentos de Teledetección Espacial*. Ediciones Rialp S.A. Madrid, 2000.
- DAVIDSSON, P. *Concept Acquisition by Autonomous Agents: Cognitive Modeling versus the Engineering Approach*. Lund University Cognitive Studies 12, ISSN 1101- 8453, Lund University, Suécia, 1992.
- DENNETT, D C. *Mind Design II: Philosophy, Psychology, Artificial Intelligence*. Oxford University Press, 1981.
- ESPINDOLA, G. M. Ajuste de parâmetros em algoritmos de segmentação de imagens por crescimento de regiões. 2006. 69 f. Dissertação (Mestrado em Sensoriamento Remoto) - Programa de pós-graduação m Sensoriamento Remoto, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2006.
- FAN, J.; YAU, D. K. Y.; ELMAGARMID, A. K.; AREF, W. G. *Automatic Image Segmentation by Integrating Color-Edge Extraction and Seeded Region Growing*. Transaction on image processing, IEEE Vol. 10, nº. 10, 2001.
- FERBER, J. *Multi-agent Systems. An Introduction to distributed artificial intelligence*. Addison-Wesley. 1999.
- FERRERO, S. B. *Evaluación de clasificaciones Máxima Verosimilitud Gaussiana y de Modas Condicionales Iteradas para imágenes ópticas multiespectrales*. PhD thesis, Facultad de Ciencia. Químicas y Naturales. Universidad Nacional de Río Cuarto, 2004.

FRANKLIN, S., GRAESSER, A. *Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents*. Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages. Springer-Verlag, 1996.

GLUTZ, J.; VICCARI, R. Linguagens de Comunicação entre agentes: Fundamentos, Padrões e Perspectivas. Anais do 23º Congresso da SBC, 3º MCIA, 2003.

GÓMEZ, A. Y.; SOTOS, A. G. *Clasificación Digital de Imágenes por Satélite*. Sistemas de Telemidia. Espanha, 2010.

GONZALEZ, R. C.; WINTZ, P. *Processamento Digital de Imagens*. New York: Addison-Wesley, 1977.

GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. Pearson Prentice Hall, New Jersey, 793p, 2002.

GREEN, W. B. *Digital Image Processing*. Ed. Van Nostrand Reinhold, USA, 1989.

HUBNER, J.; SICHTMAN, J. Organização de sistemas Multiagentes. Anais do 23º Congresso da SBC, 3º MCIA, 2003.

IBGE. Instituto Brasileiro de Geografia e Estatística. Censo demográfico do município de Angra dos Reis. Disponível em <http://www.ibge.gov.br/cidadesat/painel/painel.php>. Acessado em: abr de 2010.

JAIN, A. K. *Fundamentals of Digital Image Processing*. Prentice-Hall: Englewood Cliffs, NJ. 569p. 1989.

JENNINGS, N. R. *Agent Software*. Proceedings UNICOM Seminar on Agent Software, Londres, UK, p. 12-27, 1995.

JENSEN, J. R. *Processamento de imagem digital introdutório: Uma perspectiva de sensoriamento remoto*. Englewood Cliffs: Prentice Hall, 379 p., 1986.

KAEHLING, L. P.; ROSENSCHEIN, S. J. *Action and planning in embedded agents*. In Maes, P. editor, *Designing Autonomous Agents*. The MIT Press: Cambridge, MA, 1986.

KHOUDJIA, M.; KHANFOUF, H; MESHOUL, S. *Une Approche adaptative pour la segmentation d'images: Implémentation sur la plateforme multiagents NetLogo*. Universidade de Mentouri, Constantine, Algéria, 2001.

LIMA, A. Aplicando agentes para separar dígitos manuscritos conectados. Centro de Informática, Universidade Federal de Pernambuco, Recife, Brasil, 2005.

MASCARENHAS, N. D. A.; VELASCO, F. D. R. *Processamento Digital de Imagens*. IV Escola Brasileiro - Argentina de Informática. Termas de Rio Hondo, Argentina, 1989.

MEDINA, R.; BELLERA, J. *Bases del Procesamiento de Imágenes Médicas*. Universidad de Los Andes, Venezuela, 2004.

MELÉNDEZ, A. R. I.; ALTAMIRANO, L. R. *Estimación de Fondo y Primer Plano em Secuencias de Imágenes Para la Detección de Objetos em Movimiento*. PhD Thesis, XVI Congreso Nacional y II Congreso Internacional de Información y Computación de la ANIEI, 2003.

NETLOGO. Download do Programa Netlogo 4.1.3. Disponível em: <http://ccl.northwestern.edu/netlogo/>. Acessado em: nov de 2010.

ORTEGA, M. A. M. *Agentes Inteligentes Aplicados al Reconocimiento de Imágenes em el Campo de la Agricultura*. 2008. 96 f. Projeto (Graduação em Engenharia de Sistemas) - Faculdade de Ingenieri: Electrica, electronica, física y ciências de la computacion, Universidad Tecnologica de Pereira, Espanha, 2008.

PEARSON, D. *Image Processing*. Ed. Mc Graw Hill, UK, 1991.

AZEVEDO, L. L.; MENEZES, C. S.; MAGDALENA, B. C.; PONTAROLO, E. Exploração do ambiente orientado a agente NetLogo. Instituto de Ciências e Letras do Médio Araguaia. Universidade Federal do Espírito Santo, Vitória, ES, 2010.

PRATT, W. K. *Digital Image Processing*. Wiley Interscience, 1991.

RAO, A. S.; GEORGEFF, M. P. *BDI Agents: From Theory to Practice*. Autraliab Artificial Intelligence Institute. Melbourne, Australia, 1993.

RICHARDS, J. A; JIA, X. *Remote Sensing and Digital Image Analysis: An Introduction*. Springer-Verlag, Berlin, 1999.

RICHARDS, A. J. *Remote sensing digital image analysis*. Berlim: Springer - Verlag. 1994.

RÜTHER, H.; MARTINE, H. M.; MTALO, E. G. *Application of snakes and dynamic programming optimization technique in modeling of buildings in informal settlement areas*. ISPRS Journal of Photogrammetry and Remote Sensing, v. 56, p. 269-282, 2002.

RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 2nd Edition, Prentice Hall, 2003.

SABINS, F. F. *Remote Sensing: Principles and Interpretation*. NY, W.H Freeman, 1987.

SCALABRIN, E. E. *Conception et realization d'environnement de développement de systèmes d'agents cognitifs*. PhD Thesis. Université de Technologie de Compiègne, France, 1996.

SEEL, N. *Agent Theories and Architectures*. PhD thesis, Surrey University, Guildford, UK, 1989.

SHOHAM, Y. *Agent-oriented programming. Artificial Intelligence. Theories and Architectures*. Computer Science Department, Stanford University, Stanford, 1993.



TILTON, J.; LAWRENCE, W. *Interactive analysis of hierarchical image segmentation. In: International Geoscience and Remote Sensing Symposium IGARSS-2000*, New York. Proceedings. New York: IEEE Press, 2000.

VENTURIERI, A.; SANTOS, J. R. Técnicas de Classificação de Imagens para Análise de Cobertura Vegetal. In: ASSAD, E. D. e SANO, E. E. *Sistema de Informações Geográficas*. 2.ed., rev. e ampl. Brasília, DF: Embrapa-SPI/Embrapa-CPAC, 1998.

WALSH, S. J.; MENA, C. F.; DEHART, J. L.; FRIZZELLE, B. G. Ambientes característicos: ferramentas educacionais para a avaliação das causas e consequências do uso do solo e mudança da cobertura de Terra. Universidade da Carolina do Norte, Chapel Hill, USA, 2010.

WOOD, A. *Towards a Medium for Agent-Based Interaction*. School of Computer Science, The University of Birmingham, 1994

WOOLDRIDGE, M. *Intelligent Agents* In G. Weiss, editor: *Multiagent Systems*, The MIT Press, April, 1999.

WOOLDRIDGE, M., JENNINGS, N. R. *Intelligent Agents: Theory and Practice*. The Knowledge Engineering Review, vol. 10, n° 2, p. 115-152, 1995.

## APÊNDICE A - Pseudocódigo

```
Método classifica-edificação-por-área=====
  inicia variáveis locais
  para cada patch do ambiente (pixels da imagem) faça:
    se a cor do patch é vermelho analisa-feição-por-área (patch)
  fim-para
fim do método classifica-edificação-por-área
```

```
Método analisa-feição-por-área (p)=====
  para cada patch dentro de uma área quadrada de lado igual a raio
  e com canto superior esquerdo em p, faça:
    se a cor do patch é vermelho
      conte mais uma unidade para a área da feição
      mude a cor do patch para azul
    fim-para
  se a área da feição for igual à área padrão com tolerância de 20%
    para cada patch dentro de uma área quadrada de lado igual a raio
    e com canto superior esquerdo em p, faça:
      se a cor do patch é azul
        mude a cor do patch para verde
      fim-para
    fim-se
fim do método analisa-feição-por-área
```

```
Método classifica-edificação-por-geometria=====
  inicia variáveis locais
  para cada patch do ambiente (pixels da imagem) faça:
    se a cor do patch é vermelho analisa-feição-por-geometria (patch)
  fim-para
fim do método classifica-edificação-por-geometria
```

```
Método analisa-feição-por-geometria (p)=====
  para cada patch em uma mesma linha a partir de p, faça:
    se a cor do patch é vermelho
      conte mais uma unidade para a área da feição
      conte mais uma unidade para a largura da feição
      mude a cor do patch para azul
    senão prossiga na mesma linha até uma tolerância igual a raio
  fim-para
  se largura da feição for maior que largura máxima
  faça largura máxima igual à largura da feição
  mude de linha se altura da feição for menor que raio mais linha anterior
  com patch vermelho e execute o laço para anterior novamente
  senão faça altura máxima igual à altura da feição
  se a área da feição for igual à área padrão com tolerância de 20% e
  proporção entre altura máxima e largura máxima está
  de acordo com a proporção padrão então
    para cada patch dentro de uma área retangular de largura igual a
    largura máxima e altura igual à altura máxima
    e com canto superior esquerdo em p, faça:
      se a cor do patch é azul
        mude a cor do patch para verde
      fim-para
    fim-se
fim do método analisa-feição-por-geometria
```

## APÊNDICE B – Código Fonte

### =====Carregar Imagem Digital=====

```
to carregar-imagem
  import-pcolors-rgb "mambucaba.png"
end
```

### =====Atuação do Agente Textura=====

#### Agente de cor das vias

```
to classifica-vias-por-cor
  let px 0
  let py 0
  let cinza [60 60 60]
  let R 0
  let G 0
  let B 0

  repeat 300
  [
    repeat 300
    [
      ;print px
      ;print py
      ask patch px py
      [
        ;print pcolor
        set R item 0 pcolor
        set G item 1 pcolor
        set B item 2 pcolor
        if ((R >= vermelho-min) and (R <= vermelho-max) and (G >= verde-min) and (G <= verde-max) and (B >= azul-min)
and (B <= azul-max))
        [
          set pcolor cinza
        ]
      ]
      set px px + 1
    ]
    set px 0
    set py py - 1
  ]
end
```

#### Agente de cor da edificação

```
to classifica-edificacao-por-cor
  let px 0
  let py 0
  let vermelho [255 0 0]
  let R 0
  let G 0
  let B 0

  repeat 300
  [
```

```

repeat 300
[
;print px
;print py
ask patch px py
[
;print pcolor
set R item 0 pcolor
set G item 1 pcolor
set B item 2 pcolor
if ((R >= vermelho-min) and (R <= vermelho-max) and (G >= verde-min) and (G <= verde-max) and (B >= azul-min)
and (B <= azul-max))
[
set pcolor vermelho
]
]
set px px + 1
]
set px 0
set py py - 1
]
end

```

### Agente de cor da vegetação

to classifica-vegetacao-por-cor

```

let px 0
let py 0
let verde [0 255 0]
let R 0
let G 0
let B 0

repeat 300
[
repeat 300
[
;print px
;print py
ask patch px py
[
;print pcolor
set R item 0 pcolor
set G item 1 pcolor
set B item 2 pcolor
if ((R >= vermelho-min) and (R <= vermelho-max) and (G >= verde-min) and (G <= verde-max) and (B >= azul-min)
and (B <= azul-max))
[
set pcolor verde
]
]
set px px + 1
]
set px 0
set py py - 1
]
end

```

## Atuação do Agente Geometria=====

```

to analisa-feicao [pxorigem pyorigem]
  let verde [0 255 0]
  let azul [0 0 255]
  let px pxorigem
  let py pyorigem
  let vermelho [255 0 0]
  let deltax 1
  let deltay 1
  let proporcao-padrao 10
  let proporcao 1
  let areaf 0
  let largura 0
  let larguramaxima 0
  let alturamaxima 0
  let eixoy 300
  let eiox 300
  let acabei false
  let tevevermelho false
  let analisax false
  let analisay false

  ;set pxorigem pxorigem - raio
  while [not acabei]
  [
    ask patch px py
    [
      ifelse (pcolor = vermelho)
      [
        set pcolor azul
        set tevevermelho true
        ifelse (px < 300) [ set px px + 1 ]
        [ set analisax true ]
      ]
      [
        ifelse ((deltax < raio) and (px < 300))
        [
          ;set pcolor azul
          set deltax deltax + 1
          set px px + 1
        ]
        [ set analisax true ]
      ]
    ]
    if analisax
    [
      set analisax false
      set largura px - pxorigem
      if (largura > larguramaxima) [set larguramaxima largura]
      set px pxorigem
      set deltax 1
      ifelse (tevevermelho)
      [
        set tevevermelho false
        ifelse (py > -300) [ set py py - 1 ]
        [ set analisay true ]
      ]
    ]
  ]

```

```

    ifelse ((py > -300) and (deltay < raio))
    [
      set deltax deltax + 1
      set py py - 1
    ]
    [ set analisay true ]
  ]
  if analisay
  [
    set acabei true
    set alturamaxima pyorigem - py
  ]
]
]
]
set areaf larguramaxima * alturamaxima
ifelse (alturamaxima > larguramaxima) [ set proporcao alturamaxima / larguramaxima ]
[ set proporcao larguramaxima / alturamaxima ]
if ((areaf > 0.9 * area-padrao) and (areaf < 1.1 * area-padrao) and (proporcao <= proporcao-padrao))
[
  set py pyorigem
  set px pxorigem
  repeat alturamaxima
  [ repeat larguramaxima
    [
      ask patch px py [ set pcolor verde ]
      set px px + 1
    ]
    set py py - 1
    set px pxorigem
  ]
]

```

### Agente de geometria usando área

to classifica-edificacao-por-area

```

let px 0
let py 0
let vermelho [255 0 0]
let R 0
let G 0
let B 0

repeat 300
[
  repeat 300
  [
    ;print px
    ;print py
    ask patch px py
    [
      if (pcolor = vermelho)
      [
        ;analisa-feicao px py
      ] ]
    set px px + 1
  ]
  set px 0
  set py py - 1
]

```

```
]
end
```

### Agente de geometria usando área e proporção

```
to classifica-edificacao-por-geometria
```

```
  let px 0
  let py 0
  let vermelho [255 0 0]
  let R 0
  let G 0
  let B 0
```

```
  repeat 300
  [
    repeat 300
    [
      ;print px
      ;print py
      ask patch px py
      [
        if (pcolor = vermelho)
        [
          analisa-feicao px py
        ]
      ]
    ]
    set px px + 1
  ]
  set px 0
  set py py - 1
]
```

### Atuação do agente de segmentação=====

```
to detecta-borda
```

```
  let px 0
  let py 0
  let borda false
  let pxv 0
  let pyv 0
  let R 0
  let G 0
  let B 0
  let Rv 0
  let Gv 0
  let Bv 0
  let branco [255 255 255]
  let preto [0 0 0]
  let cor [100 255 0]
  ;let padrao pcolor at 1 -1
  let prox-padrao [0 0 0]
  ;let tot

```

```
  repeat 300
  [
```

```

ask patch px 0 [ set pcolor preto]
ask patch px -299 [ set pcolor preto]
set px px + 1
]
repeat 300
[
ask patch 0 py [ set pcolor preto]
ask patch 299 py [ set pcolor preto]
set py py - 1
]

set px 1
set py -1
repeat 299
[
repeat 299
[
ask patch px py
[
set pxv px + 1
set R item 0 pcolor
set G item 1 pcolor
set B item 2 pcolor
ask patch pxv py
[
set Rv item 0 pcolor
set Gv item 1 pcolor
set Bv item 2 pcolor
if (not (pcolor = branco) and not (pcolor = preto))
[
if ( not ( (abs (R - Rv) < limiar ) and (abs (G - Gv) < limiar ) and (abs (B - Bv) < limiar )))
[
set borda true
]
]
]
]
set pxv px - 1
ask patch pxv py
[
set Rv item 0 pcolor
set Gv item 1 pcolor
set Bv item 2 pcolor
if (not (pcolor = branco) and not (pcolor = preto))
[
if ( not ( (abs (R - Rv) < limiar ) and (abs (G - Gv) < limiar ) and (abs (B - Bv) < limiar )))
[
set borda true
]
]
]
]
set pyv py + 1
ask patch px pyv
[
set Rv item 0 pcolor
set Gv item 1 pcolor
set Bv item 2 pcolor
if (not (pcolor = branco) and not (pcolor = preto))
[

```



```

    if ( not ( (abs (R - Rv) < limiar ) and (abs (G - Gv) < limiar ) and (abs (B - Bv) < limiar )))
    [
        set borda true
    ]
]
]
ask patch pxv pyv
[
    set Rv item 0 pcolor
    set Gv item 1 pcolor
    set Bv item 2 pcolor
    if (not (pcolor = branco) and not (pcolor = preto))
    [
        if ( not ( (abs (R - Rv) < limiar ) and (abs (G - Gv) < limiar ) and (abs (B - Bv) < limiar )))
        [
            set borda true
        ]
    ]
]
set pyv py - 1
ask patch px pyv
[
    set Rv item 0 pcolor
    set Gv item 1 pcolor
    set Bv item 2 pcolor
    if (not (pcolor = branco) and not (pcolor = preto))
    [
        if ( not ( (abs (R - Rv) < limiar ) and (abs (G - Gv) < limiar ) and (abs (B - Bv) < limiar )))
        [
            set borda true
        ]
    ]
]
ask patch pxv pyv
[
    set Rv item 0 pcolor
    set Gv item 1 pcolor
    set Bv item 2 pcolor
    if (not (pcolor = branco) and not (pcolor = preto))
    [
        if ( not ( (abs (R - Rv) < limiar ) and (abs (G - Gv) < limiar ) and (abs (B - Bv) < limiar )))
        [
            set borda true
        ]
    ]
]
set pxv px + 1
ask patch pxv pyv
[
    set Rv item 0 pcolor
    set Gv item 1 pcolor
    set Bv item 2 pcolor
    if (not (pcolor = branco) and not (pcolor = preto))
    [
        if ( not ( (abs (R - Rv) < limiar ) and (abs (G - Gv) < limiar ) and (abs (B - Bv) < limiar )))
        [
            set borda true
        ]
    ]
]

```

```

    ]
  ]
  set pyv py + 1
  ask patch pxv pyv
  [
    set Rv item 0 pcolor
    set Gv item 1 pcolor
    set Bv item 2 pcolor
    if (not (pcolor = branco) and not (pcolor = preto))
    [
      if ( not ( (abs (R - Rv) < limiar ) and (abs (G - Gv) < limiar ) and (abs (B - Bv) < limiar )))
      [
        set borda true
      ]
    ]
  ]
  ifelse (borda) [ set pcolor preto ] [ set pcolor branco ]
]

set px px + 1
set borda false
]
set px 1
set py py - 1
]
end

```

### Atuação do agente de conflito=====

```

to compara-imagens
  let preto [0 0 0]
  let branco [255 255 255]
  let verde [0 255 0]
  let vermelho [255 0 0]
  let cinza [60 60 60]
  let cian [0 255 255]
  let pixels list preto preto
  let px 0
  let py 0
  let indice 0
  let R 0
  let G 0
  let B 0
  let pontos-fracos 0
  let ponto-medio 0
  let ponto-3-4 0
  let ponto-1-4 0
  set pixels lput preto pixels
  print pixels
  set pixels remove preto pixels
  print pixels

  set vermelho-min 91
  set vermelho-max 162
  set verde-min 67
  set verde-max 122
  set azul-min 50
  set azul-max 128

```

```

classifica-edificacao-por-cor
repeat 300
[
  repeat 300
  [
    ask patch px py
    [
      ;print pcolor
      set pixels lput pcolor pixels
      set pcolor branco
    ]
    set px px + 1
  ]
  set px 0
  set py py - 1
]
carregar-imagem
set vermelho-min 0
set vermelho-max 169
set verde-min 0
set verde-max 167
set azul-min 0
set azul-max 76
classifica-vegetacao-por-cor
;
set px 0
set py 0
foreach pixels
[
  ask patch px py
  [
    if (? = vermelho)
    [
      ifelse (pcolor = verde) ; aqui se resolve o conflito
      [
        set R item 0 pcolor
        set G item 1 pcolor
        set B item 2 pcolor
        set pontos-fracos 0
        set ponto-medio (vermelho-max - vermelho-min) / 2 + vermelho-min
        set ponto-3-4 ponto-medio + (ponto-medio / 2)
        set ponto-1-4 ponto-medio - (ponto-medio / 2)
        if ((R > ponto-3-4) or (R < ponto-1-4)) [ set pontos-fracos pontos-fracos + 1 ]
        set ponto-medio (verde-max - verde-min) / 2 + verde-min
        set ponto-3-4 ponto-medio + (ponto-medio / 2)
        set ponto-1-4 ponto-medio - (ponto-medio / 2)
        if ((G > ponto-3-4) or (G < ponto-1-4)) [ set pontos-fracos pontos-fracos + 1 ]
        set ponto-medio (azul-max - azul-min) / 2 + azul-min
        set ponto-3-4 ponto-medio + (ponto-medio / 2)
        set ponto-1-4 ponto-medio - (ponto-medio / 2)
        if ((B > ponto-3-4) or (B < ponto-1-4)) [ set pontos-fracos pontos-fracos + 1 ]
        if (pontos-fracos > 1) [ set pcolor vermelho ]
      ]
    ]
    [ set pcolor vermelho ]
  ]
  set pixels replace-item indice pixels pcolor
  set indice indice + 1
]

```

```

set px px + 1
if (px = 300)
[
  set px 0
  set py py - 1
]
]
carregar-imagem
set vermelho-min 114
set vermelho-max 234
set verde-min 96
set verde-max 219
set azul-min 104
set azul-max 219
classifica-vias-por-cor
;
set px 0
set py 0

foreach pixels
[
  ask patch px py
  [
    if ((? = vermelho) or (? = verde))
    [
      ifelse (pcolor = cinza) ; aqui se resolve o conflito
      [
        set R item 0 pcolor
        set G item 1 pcolor
        set B item 2 pcolor
        set pontos-fracos 0
        set ponto-medio (vermelho-max - vermelho-min) / 2 + vermelho-min
        set ponto-3-4 ponto-medio + (ponto-medio / 2)
        set ponto-1-4 ponto-medio - (ponto-medio / 2)
        if ((R > ponto-3-4) or (R < ponto-1-4)) [ set pontos-fracos pontos-fracos + 1 ]
        set ponto-medio (verde-max - verde-min) / 2 + verde-min
        set ponto-3-4 ponto-medio + (ponto-medio / 2)
        set ponto-1-4 ponto-medio - (ponto-medio / 2)
        if ((G > ponto-3-4) or (G < ponto-1-4)) [ set pontos-fracos pontos-fracos + 1 ]
        set ponto-medio (azul-max - azul-min) / 2 + azul-min
        set ponto-3-4 ponto-medio + (ponto-medio / 2)
        set ponto-1-4 ponto-medio - (ponto-medio / 2)
        if ((B > ponto-3-4) or (B < ponto-1-4)) [ set pontos-fracos pontos-fracos + 1 ]
        if ((pontos-fracos > 1) and (? = vermelho)) [ set pcolor vermelho ]
        if ((pontos-fracos > 1) and (? = verde)) [ set pcolor verde ]
      ]
      [ set pcolor ? ]
    ]
  ]
  ;set pixels replace-item indice pixels pcolor
  ;set indice indice + 1
]
set px px + 1
if (px = 300)
[
  set px 0
  set py py - 1
]
]

```