

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Engenharia de Computação e Automação Industrial

**Sistema de Recuperação de Imagens Baseado na
Teoria Computacional das Percepções e em
Linguagens Formais Fuzzy**

AUTOR: MATEUS GUILHERME FUINI

ORIENTADOR: PROF. DR. FERNANDO ANTONIO CAMPOS GOMIDE

Dissertação de Mestrado apresentada à Faculdade Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, como parte dos requisitos exigidos para obtenção do título de **Mestre em Engenharia Elétrica**.

Área de Concentração: Engenharia de Computação.

Banca Examinadora:

PROF. DR. EDUARDO MASSAD (DIM/FM/USP)

PROF. DR. FERNANDO JOSÉ VON ZUBEN (DCA/FEEC/UNICAMP)

PROF. DR. RICARDO RIBEIRO GUDWIN (DCA/FEEC/UNICAMP)

Campinas, SP
Junho/2006

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

F955s Fuini, Mateus Guilherme
Sistema de recuperação de imagens baseada na teoria
computacional das percepções e em linguagens formais fuzzy
/ Mateus Guilherme Fuini. --Campinas, SP: [s.n.], 2006.

Orientador: Fernando Antônio Campos Gomide
Dissertação (Mestrado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Processamento de imagens – Técnicas digitais. 2.
Linguagens formais. 3. Lógica difusa. 4. Visão por
computador. 5. Algoritmos. I. Gomide, Fernando Antônio
Campos. II. Universidade Estadual de Campinas. Faculdade de
Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Image retrieval system based in computacional theory perceptions and
fuzzy formal language.

Palavras-chave em Inglês: Digital image processing, Formal language, Fuzzy logic,
Computer vision, Algorithms

Área de concentração: Engenharia da Computação

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Fernando Antônio Campos Gomide, Eduardo Massad, Fernando
José Von Zuben e Ricardo Ribeiro Gudwin.


Data da defesa: 05/06/2006

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Mateus Guilherme Fuini

Data da Defesa: 5 de junho de 2006

Título da Tese: "Sistema de Recuperação de Imagens Baseada na Teoria Computacional das Percepções e em Linguagens Formais Fuzzy."

Prof. Dr. Fernando Antônio Campos Gomide (Presidente): _____ 

Prof. Dr. Eduardo Massad: _____ 

Prof. Dr. Fernando José Von Zuben: _____ 

Prof. Dr. Ricardo Ribeiro Gudwin: _____ 

RESUMO

Neste trabalho utilizam-se as teorias de Linguagem Formal Nebulosa e da Computacional das Percepções de Zadeh para definir buscas em uma base de dados gráfica. A descrição dos elementos gráficos a serem identificados é codificada por meio de sentenças aceitas por uma gramática nebulosa e definida sobre um conjunto de símbolos gráficos terminais reconhecidos por rotinas computacionais específicas. Esses símbolos terminais rotulam a imagem a ser pesquisada. A teoria da Percepção Computacional é usada para permitir que o usuário defina as relações espaciais a serem compartilhadas pelos elementos gráficos na cena a ser pesquisada. Os resultados obtidos com buscas realizadas em uma base de dados gráfica com 22000 desenhos mostram que o sistema proposto fornece uma alternativa interessante para solução de buscas em bancos de dados visuais.

Palavras-chave: 1. Processamento Digital de imagens. 2. Linguagens formais. 3. Lógica difusa 4. Visão por computador 5. Algoritmos

ABSTRACT

In this work, Fuzzy Formal Language techniques and Zadeh's Computational Theory of Perceptions are used to allow the user to query graphic data bases. The description of the graphic elements to be searched is encoded by means of fuzzy sentences accepted by a fuzzy grammar defined over a set of graphic primitives recognized by specific computational routines aimed to label different primitive graphic components of a given image. The Computational Theory of Perceptions is used to allow the user to specify the required spatial relations to be shared by the selected in the graphic scenes to be selected. The results obtained by querying a 22000 graphic scene data base support the claim that our approach provides a interesting solution for querying visual data bases.

Keywords: 1. Digital Image Processing. 2. Formal Language. 3. Fuzzy Logic 4. Computer Vision 5. Algorithms

AGRADECIMENTOS

Agradeço em primeiro lugar, a Deus, pela realização deste sonho e por colocar pessoas competentes e dedicadas em meu caminho e que me ajudaram na realização deste trabalho.

Aos meus pais, que nunca mediram esforços para que eu pudesse realizar este sonho. Ao meu irmão, pelo apoio e incentivo.

Aos Professores Dr. Fernando Gomide e Dr. Armando Freitas da Rocha pelas orientações e sugestões neste trabalho.

Ao Prof. Dr. Fernando Von Zuben pela dedicação com que sempre me recebeu e pelas sugestões e comentários sobre o trabalho.

Ao Prof. Dr. Ricardo Gudwin pela participação na banca examinadora e pelas sugestões e correções proposta que enriqueceram este trabalho.

Ao amigo Marcos Paulo Rebello, pela grande ajuda na etapa de implementação deste projeto.

Aos amigos e companheiros Fábio Lucchini e Ednilson Rodella pelo suporte em várias etapas deste projeto.

À amiga Cássia Medea, pelo apoio e pelas ilustrações deste projeto.

Finalmente, meus sinceros agradecimentos à FAPESP, pelo financiamento deste projeto (02/04981-1).

Sumário

1	INTRODUÇÃO	1
1.1	MOTIVAÇÃO	1
1.2	OBJETIVO DA PESQUISA.....	1
1.3	METODOLOGIA DO DESENVOLVIMENTO DA PESQUISA	2
1.4	ESTRUTURA DO TRABALHO	2
2	CONCEITOS E TEORIAS.....	5
2.1	SISTEMA DE RECUPERAÇÃO DE IMAGEM	5
2.2	SISTEMAS DE RECUPERAÇÃO DE IMAGENS ATUAIS	6
2.3	SISTEMA VISUAL HUMANO	8
2.3.1	<i>Recomposição de uma cena</i>	<i>10</i>
2.4	LINGUAGENS FORMAIS CLÁSSICAS E NEBULOSAS.....	10
2.4.1	<i>Introdução.....</i>	<i>11</i>
2.4.2	<i>Gramática Formal Fuzzy</i>	<i>16</i>
2.4.3	<i>Teoria Computacional das Percepções.....</i>	<i>23</i>
2.4.4	<i>Gramática e Percepção Computacional do Sistema</i>	<i>24</i>
3	O SISTEMA SENSOR V2	29
3.1	PRIMEIRAS VERSÕES.....	29
3.2	VERSÃO ATUAL.....	34
3.2.1	<i>Inserção e compilação das imagens da base de dados</i>	<i>34</i>
3.2.2	<i>Construindo uma sentença.....</i>	<i>36</i>
3.2.3	<i>O processamento das buscas e os resultados encontrados</i>	<i>37</i>
4	ESTRUTURA DO SISTEMA SENSOR V2.....	39
4.1	ESTRUTURA DO SISTEMA	39
4.1.1	<i>Diagramas do sistema.....</i>	<i>40</i>
4.2	EXTRAÇÃO DE INFORMAÇÕES DAS IMAGENS	44
4.2.1	<i>Filtros de Imagem</i>	<i>44</i>
5	RESULTADOS	45
5.1	COMPARAÇÕES.....	58
5.1.1	<i>Comparação de eficiência e velocidade.....</i>	<i>62</i>
6	CONCLUSÕES.....	69
6.1	SUGESTÕES DE TRABALHOS FUTUROS.....	70
7	BIBLIOGRAFIA	73

APÊNDICE A	83
<i>Algoritmos</i>	93
<i>Arquiteturas</i>	93
<i>Primitivas de Consultas sobre Imagens</i>	93
APÊNDICE B	95
FILTROS UTILIZADOS DURANTE O DESENVOLVIMENTO DO SISTEMA.....	97
EXTRAÇÃO DE ÁREAS.....	99
<i>Contorno</i>	99
APÊNDICE C	103

LISTA DE FIGURAS

Figura 2-1 - Um esquema de sistema de recuperação e armazenagem de imagem	6
Figura 2-2 – As distintas imagens da mesma cena geradas pela retina.....	8
Figura 2-3 – A análise cortical de linhas e ângulo	9
Figura 2-4 – A análise cortical para identificação do objeto.....	10
Figura 2-5 – A análise cortical do movimento e localização espacial	10
Figura 2-6 - Grafo de uma sentença da língua portuguesa.....	14
Figura 2-7 - Grafo da personagem Juca	26
Figura 2-8 -Grafo S_{juca}	27
Figura 2-9 - Grafo do objeto Laura	28
Figura 2-10 - Grafo de uma cena	28
Figura 3-1 - Organização computacional dos filtros.....	29
Figura 3-2 - Estrutura de agentes utilizados na simulação do sistema visual humano.	30
Figura 3-3 - Objetos para descrever uma cena.....	33
Figura 3-4 - Cena Prototípica Média.....	34
Figura 3-5 – Imagem Inserida no sistema	35
Figura 3-6 – Resultado do processamento. Áreas e rotulação, cada cor representando uma área.	35
Figura 3-7 – Imagem somente com os contornos e em destaque os pontos significativos da cabeça do personagem Juca e a respectiva sentença desta área	36
Figura 3-8 - Montagem de uma sentença para o objeto Juca (Traços maiores).....	37

Figura 3-9 – Sentença do objeto Juca.....	37
Figura 3-10 – Imagem em que o sistema reconheceu o objeto Juca	38
Figura 3-11 - Objeto reconhecido na Imagem	38
Figura 4-1 - Diagrama de componentes	40
Figura 4-2 - Diagrama de casos de uso	41
Figura 4-3 - Diagrama de Atividades - Inserindo uma imagem.....	41
Figura 4-4 - Diagrama de Atividades - Criação de Objetos	42
Figura 4-5 - Diagrama de Atividades - Pesquisa das sentenças	43
Figura 5-1- Personagem procurada - Juca.....	48
Figura 5-2 - Imagem Recuperada mesmo com o objeto parcialmente ocluído.....	49
Figura 5-3 Imagem falsa recuperada devido a semelhança dos traços do cabelo	49
Figura 5-4 - Imagem recuperada pelo cabelo do personagem	49
Figura 5-5 Imagem falsa recuperada pelas listras vermelhas e amarelas.....	49
Figura 5-6 - Grafo da retroalimentação do objeto Juca.....	49
Figura 5-7 – Resultado correto com 100% de semelhança	50
Figura 5-8 – Resultado falso positivo com índice 53%	50
Figura 5-9 – Falso negativo com 48% de semelhança	50
Figura 5-10 - Resultado positivo com índice de 90%	51
Figura 5-11 – Resultado falso positivo com índice de 60%.....	51
Figura 5-12 – Um dos resultados positivos com 84% de semelhança	52

Figura 5-13 – Resultado falso positivo com índice de 52%.....	52
Figura 5-14 – Resultado falso negativo com índice de 49%.....	52
Figura 5-15 Personagem procurada - Laura.....	52
Figura 5-16 - Grafo do objeto de busca para Laura	53
Figura 5-17 – Resposta correta com 100% de semelhança.....	53
Figura 5-18 - Resultado falso positivo com índice de 70%.	53
Figura 5-19 – Resultado falso negativo com índice de 26%.....	53
Figura 5-20 - Personagem Vovô	54
Figura 5-21 - Grafo da busca do personagem Vovô	56
Figura 5-22 - Algumas figuras de árvores presente na base de dados	57
Figura 5-23 - Grafo do objeto de busca árvore	58
Figura 5-24 - Imagem a processar.....	60
Figura 5-25 – Contorno fechado e a curvatura da forma com os pontos de cruzamento.....	60
Figura 5-26 – Evolução da forma: contorno após 24 iterações.....	60
Figura 5-27 – Comportamento da curvatura após 24 iterações.....	60
Figura 5-28 - Evolução da forma: contorno após 35 iterações	60
Figura 5-29 – Comportamento da curvatura após 35 iterações.....	60
Figura 5-30 - Evolução da forma: contorno após 60 iterações	61
Figura 5-31 – Fim do processamento: Todos os picos CSS.....	61
Figura 5-32 – Imagem a rotulada após a execução do algoritmo water-filling.....	62

Figura 5-33 – Alguns resultados do objeto Juca	63
Figura 5-34 - Alguns resultados do objeto Árvore.....	64
Figura 5-35 – Alguns resultados do objeto Laura	64
Figura 5-36 - Precisão média das buscas dos 3 sistemas	66
Figura 5-37 - Comparação de tempo entre os 3 sistemas.....	67
Figura A-0-1 - Arquitetura do sistema QBIC.....	85
Figura A-0-2 - Arquitetura do VisualSEEK.....	87

LISTA DE TABELAS

Tabela 5-1 - Tabelas de resultados	47
Tabela 5-2 - Resultados da busca do personagem Juca	48
Tabela 5-3 - Resultados da busca do personagem Laura	52
Tabela 5-4 - Resultados da busca do personagem Vovô.....	55
Tabela 5-5 - Resultados da busca do personagem Árvore	57

1 Introdução

1.1 Motivação

Um algoritmo para aprendizagem e reconhecimento de padrões visuais, chamado *algoritmo de aprendizagem por moldes*, foi desenvolvido em nosso laboratório (Serapião, 1996, Lucchini, 2001) baseado nos conhecimentos atuais sobre o funcionamento do sistema visual natural. Os padrões visuais são contornos de imagem extraídos pelo sistema visual artificial como sentenças que obedecem a uma gramática nebulosa. O aprendizado de padrões correspondentes a objetos definidos se faz obtendo-se sentenças prototípicas ou “promediadas” de cada objeto de interesse.

Para descrever objetos e cenas, Rodella (2002) baseou-se na teoria computacional das percepções de Zadeh(2001), a qual propõe que é possível realizar uma grande variedade de tarefas físicas e mentais sem a necessidade de medições ou cálculos precisos. Os valores dos atributos são granulados e cada grânulo reúne um grupo de valores (pontos, objetos) por similaridade, proximidade ou função, que, por exemplo, os grânulos de uma distância poderiam ser definidos como *muito perto, perto, médio, longe, muito longe*, dentre outros. Pressupõe-se que as percepções humanas são organizadas a partir de F-grânulos obtidos como processamento de uma Linguagem de Restrições Generalizadas (LRG), de modo que a propagação das restrições tem como meta responder a uma determinada questão (Zadeh, 2001).

1.2 Objetivo da Pesquisa

O propósito do projeto de pesquisa que levou a execução deste trabalho foi implementar um sistema de visão artificial que simule o sistema visual natural, para buscas em base de dados de desenhos, utilizando e ampliando para isso, as técnicas de rotulação de imagens propostas por Serapião, 1996; as técnicas de reconhecimento de imagens implementadas por Lucchini, 2001 e as técnicas de descrição de objetos e cenas utilizadas por Rodella, 2002.

Portanto, desenvolveu-se um módulo para o processamento digital de imagens (PDI) e outro para a busca das imagens que satisfaçam certo critério de pesquisa (sentença).

As principais funções, disponíveis na literatura, para filtragem, segmentação, seleção de características, reconhecimento e interpretação de imagens foram implementadas no módulo de PDI. No módulo de busca, foram implementados as interfaces de uso e os métodos nebulosos para tomada de decisões.

1.3 Metodologia do desenvolvimento da pesquisa

O estudo em questão foi dividido em diversas etapas, que juntas, integram o trabalho desenvolvido. Assim, o estudo teve início a partir das sugestões de trabalhos futuro citados em Serapião(1996), Lucchini(2001), Rodella(2002).

O primeiro passo foi realizar um levantamento bibliográfico acerca dos conceitos, dos desenvolvimentos e das aplicações em reconhecimento de padrões, indexação e recuperação de imagens em sistemas artificiais, detalhados no Apêndice A, e dos mecanismos naturais de como o cérebro realiza essas tarefas.

Uma vez conhecidas essas técnicas, partiu-se para o desenvolvimento de um sistema de recuperação de imagens. O sistema foi dividido em duas etapas: a primeira foi o desenvolvimento de um módulo para o processamento digital das imagens para extrair as informações e realizar indexações; e a segunda foi o desenvolvimento da interface para a montagem das sentenças de busca e métodos nebulosos para tomada de decisões.

O trabalho é encerrado apresentando as comparações com outros sistemas e as conclusões obtidas dos experimentos realizados.

1.4 Estrutura do Trabalho

Este trabalho está estruturado em sete capítulos:

A motivação e os objetivos, assim como a estrutura do trabalho realizado são descrito no capítulo 1. No segundo capítulo, introduzem-se conceitos importantes para o entendimento do trabalho desenvolvido.

No terceiro capítulo, é dada uma visão geral do funcionamento do sistema desenvolvido.

A descrição da estrutura básica do sistema de recuperação gráfica de imagens e as ferramentas implementadas para extrair características das imagens são mostradas no capítulo 4.

Os resultados e as comparações são apresentados no capítulo 5, e no capítulo 6 as conclusões do trabalho.

No Apêndice A são discutidos vários sistemas de recuperação de imagens disponíveis na literatura. No Apêndice B, os filtros de imagens implementados e utilizados no sistema e, no Apêndice C alguns algoritmos implementados no sistema.

2 Conceitos e Teorias

Neste capítulo serão apresentados alguns conceitos e teorias sobre os Sistemas de Recuperação de Imagens, o Sistema Visual Humano e a Gramática Formal Fuzzy.

2.1 Sistema de Recuperação de Imagem

Devido ao advento da tecnologia de multimídia, os estudos mais recentes estão preocupados em encontrar uma forma adequada para a indexação e recuperação de imagens, semelhante ao que hoje é feito com dados alfanuméricos (Souza, 1999).

Nos trabalhos da década de 90, a descrição de imagens era feita com anotações de textos gerados pelos usuários. O título e/ou a legenda das imagens e mais alguma descrição adicional eram associados e armazenados com as imagens, conforme os trabalhos descritos em Leung (1990), Bordogna (1990) e Chakravarthy (1994). Devido ao esforço das anotações manuais e às diferentes interpretações dos usuários essa abordagem é restrita.

Conforme Souza (1999), um sistema de recuperação de imagem deve operar sobre uma coleção de imagens e, em resposta a uma solicitação (*requisição*), apresentar "imagens relevantes" segundo os critérios estabelecidos. A "*requisição*" é definida pelo usuário. Ele pode fornecer, por exemplo, as características que as imagens recuperadas deveriam conter ou mesmo um pedaço de uma imagem ou, ainda, uma imagem inteira, e o sistema deve então encontrar imagens que podem ser visualmente semelhantes.

Relevância está associada à *semântica*. Para encontrar todas as figuras de cartões de Natal em uma determinada base de dados temos que associar *semanticamente*, Papai Noel, árvore de Natal, rena e neve à cartões de natal. Para codificarmos essas *informações semânticas* precisamos utilizar técnicas de segmentação, reconhecimento e extração automática de característica das imagens.

Uma abordagem proposta freqüentemente para solucionar o problema da relevância vem da observação de que, em muitos casos, os atributos associados a uma imagem, quando usados

junto com alguma informação fornecida pelo usuário, capturam algum tipo de informação semântica que é desejável (Souza, 1999).

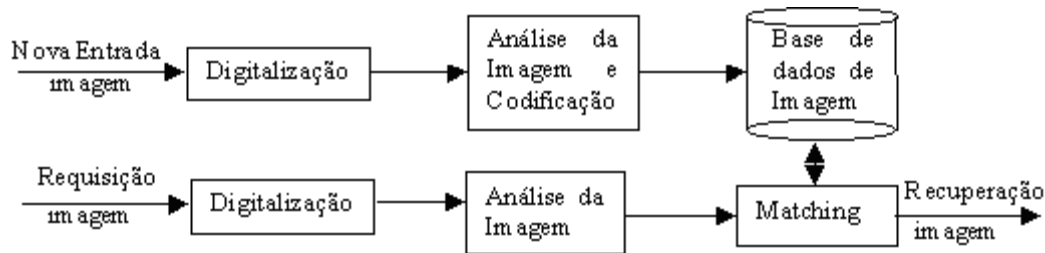


Figura 2-1 - Um esquema de sistema de recuperação e armazenagem de imagem

Os pesquisadores têm desenvolvido técnicas que permitam armazenar e recuperar imagens baseadas em seu conteúdo. A Figura 2-1(Mandal et al, 1996) é o diagrama de um sistema de indexação e recuperação de imagens.

Neste contexto, existem dois grupos de pesquisadores. O primeiro usa os algoritmos de visão computacional para gerar a descrição do objeto e armazená-lo em uma forma textual. Essas anotações são transferidas, então, a um sistema de recuperação. O segundo tipo dá ênfase a objetos semelhantes baseado em forma, cor e textura (Souza, 1999).

No apêndice A, encontra-se um estudo dos principais sistemas de recuperação de imagens disponíveis na literatura.

2.2 Sistemas de recuperação de imagens atuais

Há duas preocupações fundamentais nas construções dos sistemas de recuperação de imagens: rapidez e eficiência. Com esse objetivo os sistemas têm evoluído, desenvolvendo técnicas que melhor integram as características visuais das imagens. Para bases de dados pequenas (na ordem de centenas de imagens) a maioria das técnicas têm mostrado bons desempenhos. Conforme o tamanho da base de dados cresce, essas técnicas não são satisfatórias na prática. Além disto, estas mesmas técnicas têm sido testadas em meios controlados. Conseqüentemente, a robustez em relação aos ruídos e distorções nos ambientes da vida real ainda não foram demonstrados.

Problemas muito difíceis em sistemas de imagens permanecem ainda em investigação. Existe uma proliferação explosiva das imagens digitais sem restrições na forma de gráficos e vídeos devido ao melhoramento do acesso às tecnologias computacionais e à maior aceitabilidade do mundo Web como um meio viável para comunicação, publicação e propaganda. Isso tem criado uma necessidade de ferramentas eficientes e efetivas para catalogar, indexar, gerenciar, comprimir e buscar informações visuais sem restrições. Uma grande lacuna permanece entre a habilidade computacional em analisar imagens e vídeos em um nível de características (cor, textura, forma) e a inabilidade em um nível semântico (cenas, pessoas, valores artísticos). O aperfeiçoamento de tecnologias para preencher esta lacuna pode melhorar sistemas já existentes ou criar novas abordagens (Souza, 1999).

O maior problema de um sistema de indexação e recuperação de imagem é a determinação de um *conjunto de características* que possa fornecer um bom desempenho na classificação das imagens para uma recuperação adequada. A esse *conjunto de características* devem-se os diferentes tipos de sistemas encontrados na literatura. Assim, ainda existe a necessidade do desenvolvimento de ferramentas que possibilitem ou contribuam para o desenvolvimento de um sistema de indexação e recuperação de imagens.

2.3 Sistema Visual Humano

Diversos trabalhos têm sido desenvolvidos em nosso laboratório para criar uma teoria de sistema visual artificial e natural a partir do conhecimento sobre a visão biológica (Rocha 1997,1999,2000; Alegre 1993a,1993b).

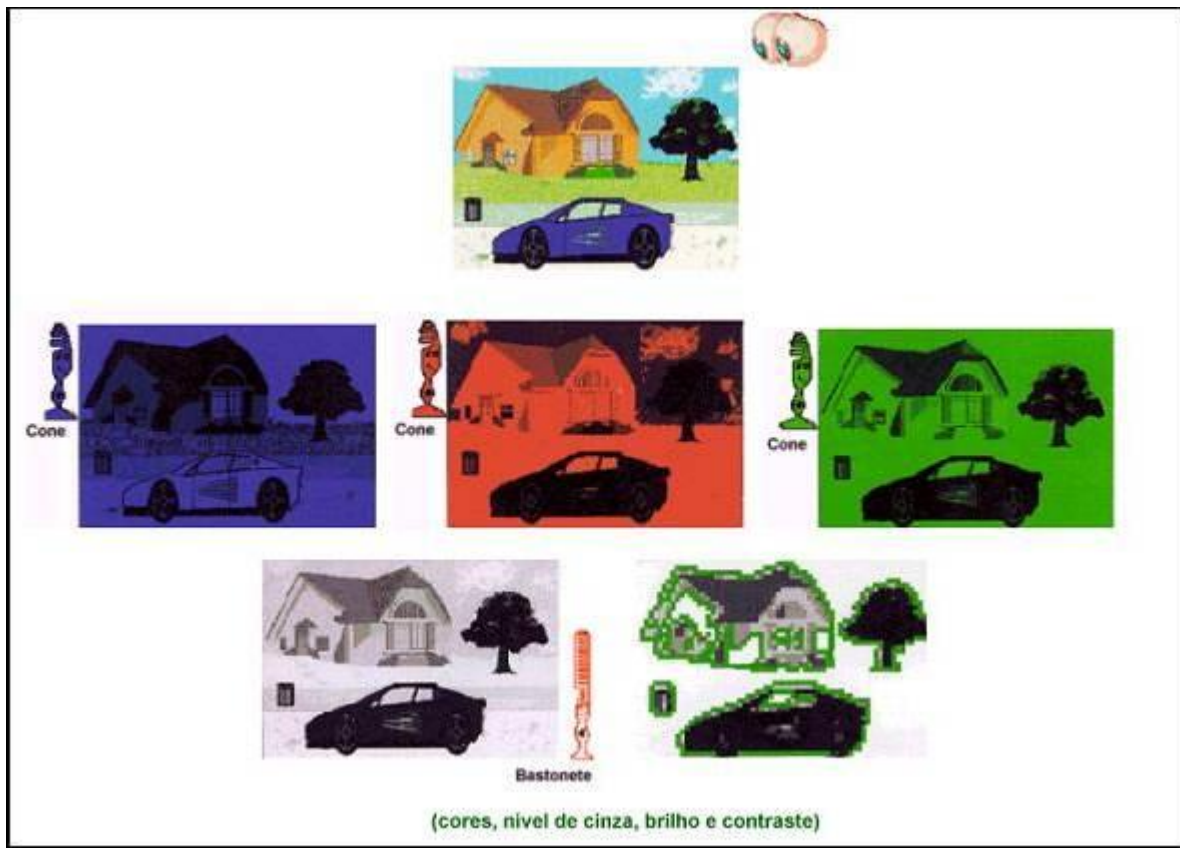


Figura 2-2 – As distintas imagens da mesma cena geradas pela retina

As ondas luminosas emitidas pelos objetos que vemos, no mundo que nos cerca, criam uma imagem desse objeto sobre a retina, que realiza um processamento complexo da imagem visual, estimulando dois tipos diferentes de receptores: os bastonetes e os cones. Os cones são sensíveis a três cores: verde, vermelho e azul, gerando assim três imagens (Figura 2-2), enquanto que os bastonetes são ativados por ondas luminosas de qualquer frequência, medindo assim a luminância da cena observada e gerando uma imagem em tons de cinza (Figura 2-2). Com isso geram-se tipos distintos de informação para as outras áreas cerebrais (Rocha, 1999).

Essas informações ativam áreas visuais primárias cuja função é extrair características básicas de imagem, tais como direções predominantes de linhas; composição de cores, análise de direções e velocidades de movimentação, etc., que serão utilizadas no reconhecimento dos elementos que compõem a cena a ser analisada (Rocha, 1999).

As áreas visuais secundárias especializam-se em analisar características particulares da imagem registrada na retina. Identificadas as principais linhas que compõem a imagem e com essa informação identificam-se os ângulos formados por essas linhas; outras áreas, ainda, analisam as discontinuidades existentes nessas linhas (Figura 2-3).

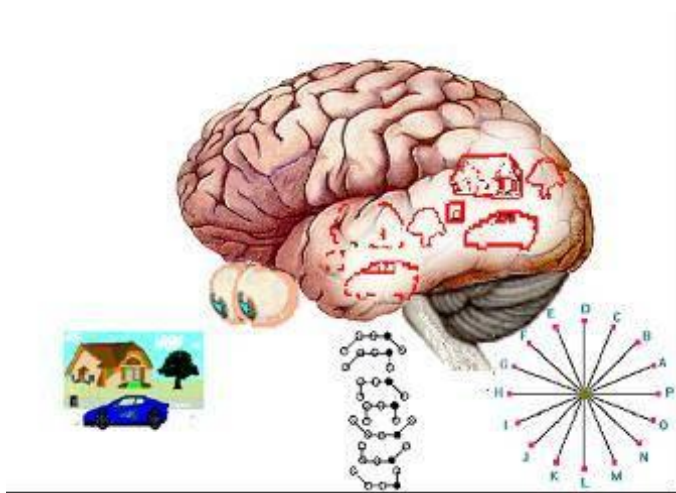


Figura 2-3 – A análise cortical de linhas e ângulo

Na sequência, o resultado do processamento nas áreas visuais secundárias é transferido para áreas de associação, distribuídas nos lobos temporal e parietal. O papel dessas áreas de associação é reconhecer os principais elementos que compõem a cena, a partir de suas características básicas. Cada elemento é reconhecido pelas relações específicas partilhadas por um conjunto de características definidas. Quando se aprende a reconhecer um objeto, se aprende a identificar esse conjunto de características específicas e as relações estabelecidas entre elas.

2.3.1 Recomposição de uma cena

A segunda etapa do processamento sensorial envolve a recomposição da cena final a partir dos elementos individualmente reconhecidos e das relações partilhadas por eles na imagem observada e identificadas pelo cérebro.

É postulado que a identificação dos objetos (“O quê”) é um trabalho preferencial do hemisfério esquerdo(Figura 2-4), enquanto que a análise das relações espaciais (“Onde”) fica mais a cargo do hemisfério cerebral direito(Figura 2-5). Essa análise espacial envolve também a composição e análise da coerência dos movimentos associados a cada um dos objetos que compõem a cena visualizada.

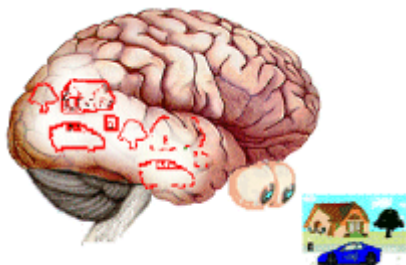


Figura 2-4 – A análise cortical para identificação do objeto

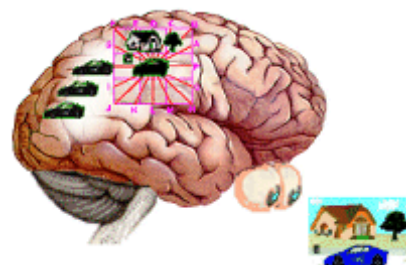


Figura 2-5 – A análise cortical do movimento e localização espacial

2.4 Linguagens Formais Clássicas e Nebulosas

Nesta seção, será apresentada a formalização do conceito de gramática formal nebulosa inicialmente desenvolvida por Rocha e outros (Mizumoto et al., 1973; Negoita e Ralescu, 1975; Serapião, Rocha e Rebello, 1996; Rocha, Françoso, e Balduino, 1980). Primeiro é apresentada uma extensão das linguagens formais clássicas para o suporte aos conceitos da gramática formal nebulosa.

2.4.1 Introdução

Uma linguagem é qualquer conjunto ou subconjunto de sentenças sobre um conjunto finito não vazio de símbolos, chamado alfabeto.

Uma gramática, de maneira informal, pode ser definida como sendo:

a) Um sistema gerador de linguagens;

b) Um sistema de reescrita;

c) Uma maneira finita de descrever (representar) uma linguagem;

d) Um dispositivo formal usado para especificar de maneira finita e precisa uma linguagem potencialmente infinita.

Exemplo intuitivo de uma Gramática (um subconjunto da gramática da língua portuguesa):

Considere a seguinte notação

$\langle \dots \rangle$: categoria sintática ou gramatical;

$::=$: definido por

$|$: ou (alternativa)

$\alpha ::= \beta$: regra de sintaxe (ou regra gramatical ou regra de produção).

Temos:

$\langle \text{sentença} \rangle ::= \langle \text{sujeito} \rangle \langle \text{predicado} \rangle$

$\langle \text{sujeito} \rangle ::= \langle \text{substantivo} \rangle | \langle \text{artigo} \rangle \langle \text{substantivo} \rangle | \langle \text{artigo} \rangle \langle \text{substantivo} \rangle \langle \text{adjetivo} \rangle$

$\langle \text{predicado} \rangle ::= \langle \text{verbo} \rangle \langle \text{objeto} \rangle | \langle \text{verbo} \rangle$

$\langle \text{substantivo} \rangle ::= \text{João} | \text{Maria} | \text{cachorro} | \text{livro} | \text{pão}$

$\langle \text{artigo} \rangle ::= \text{o} | \text{a}$

<adjetivo> ::= doce | bom | bela

<verbo> ::= morde | lê | olha

<objeto> ::= <substantivo> | <artigo> <substantivo> | <artigo> <adjetivo> <substantivo>

Formalmente definimos uma gramática G como sendo um sistema formal constituído de quatro elementos

$G = (V, T, P, S)$ onde:

V – É um conjunto finito de símbolos denominados não-terminais. Esses símbolos também são denominados meta-variáveis, ou seja, são os símbolos utilizados na descrição da linguagem.

T – É um conjunto finito de símbolos denominados terminais. São os símbolos da linguagem propriamente ditos, ou seja, os símbolos que podem ser usados na formação das sentenças da linguagem.

P - conjunto finito de pares, denominados regras de produção tal que o primeiro componente é palavra de $(V \cup T)^+$ e o segundo componente é palavra de $(V \cup T)^*$;¹

Uma regra de produção (α, β) é representada por $\alpha \rightarrow \beta$.

S – É o símbolo inicial da gramática; deve pertencer a V. O símbolo inicial de uma gramática é o não-terminal a partir do qual as sentenças de uma linguagem serão geradas.

$$\begin{aligned} V \cap T &= \emptyset \\ \Sigma &= V \cup T \\ S &\in V \\ P &= \{ \alpha \rightarrow \beta \mid \alpha \in \Sigma^+ \text{ e } \beta \in \Sigma^* \} \end{aligned}$$

¹ Neste trabalho utilizaremos a notação S^* para representar o conjunto de todas as cadeias formadas com elementos de S, incluindo-se a cadeia vazia. A notação Σ^+ é equivalente a essa, excluindo-se a cadeia vazia.

Conforme Rocha(1997) pode-se dizer que a aplicação de uma regra implica em três operações:

Comparação: o lado esquerdo da provável regra de reescrita é comparado com os símbolos da cadeia do lado direito a ser processada. Se essa comparação for bem sucedida, então:

Reescrita: a subcadeia selecionada do lado esquerdo é substituída pelo lado direito da regra de reescrita aceita. Finalmente:

Avaliação: valor booleano que informa se houve ou não a possibilidade de reescrita, ou seja, se o passo de reescrita foi executado.

Exemplo:

Formalizando o subconjunto da gramática da língua portuguesa apresentado acima, teríamos:

$G_{portugues} = (V, T, P, S)$, onde:

$V = \{ \langle \text{sentença} \rangle, \langle \text{sujeito} \rangle, \langle \text{predicado} \rangle, \langle \text{substantivo} \rangle, \langle \text{artigo} \rangle, \langle \text{adjetivo} \rangle, \langle \text{verbo} \rangle, \langle \text{objeto} \rangle \}$

$T = \{ \text{João, Maria, cachorro, livro, pão, o, a, pequeno, bom, bela, morde, lê, olha} \}$

$P = \{ \langle \text{sentença} \rangle \rightarrow \langle \text{sujeito} \rangle \langle \text{predicado} \rangle, \langle \text{sujeito} \rangle \rightarrow \langle \text{substantivo} \rangle | \langle \text{artigo} \rangle \langle \text{substantivo} \rangle | \langle \text{artigo} \rangle \langle \text{adjetivo} \rangle \langle \text{substantivo} \rangle \langle \text{predicado} \rangle \rightarrow \langle \text{verbo} \rangle \langle \text{objeto} \rangle | \langle \text{verbo} \rangle \langle \text{substantivo} \rangle \rightarrow \text{João} | \text{Maria} | \text{cachorro} | \text{livro} | \text{pão} \langle \text{artigo} \rangle \rightarrow \text{o} | \text{a} \langle \text{adjetivo} \rangle \rightarrow \text{pequeno} | \text{bom} | \text{bela} \langle \text{verbo} \rangle \rightarrow \text{morde} | \text{lê} | \text{olha} \langle \text{objeto} \rangle \rightarrow \langle \text{substantivo} \rangle | \langle \text{artigo} \rangle \langle \text{substantivo} \rangle | \langle \text{artigo} \rangle \langle \text{substantivo} \rangle \langle \text{adjetivo} \rangle \}$

$S = \langle \text{sentença} \rangle$

Fazendo uma analogia entre o exemplo intuitivo e a noção formal de gramática, constatamos que:

V – são as categorias sintáticas ou gramaticais;

T – são as palavras utilizadas como símbolos da linguagem;

P – são as regras sintáticas (ou gramaticais);

S - é a categoria gramatical que sintetiza o que será produzido (gerado) pela gramática.

Assim, por exemplo, a sentença *João morde o pão doce* é uma fórmula bem formada (fbf), pois obedece a gramática $G_{\text{portugues}}$. A Figura 2-6 ilustra o grafo desta sentença.

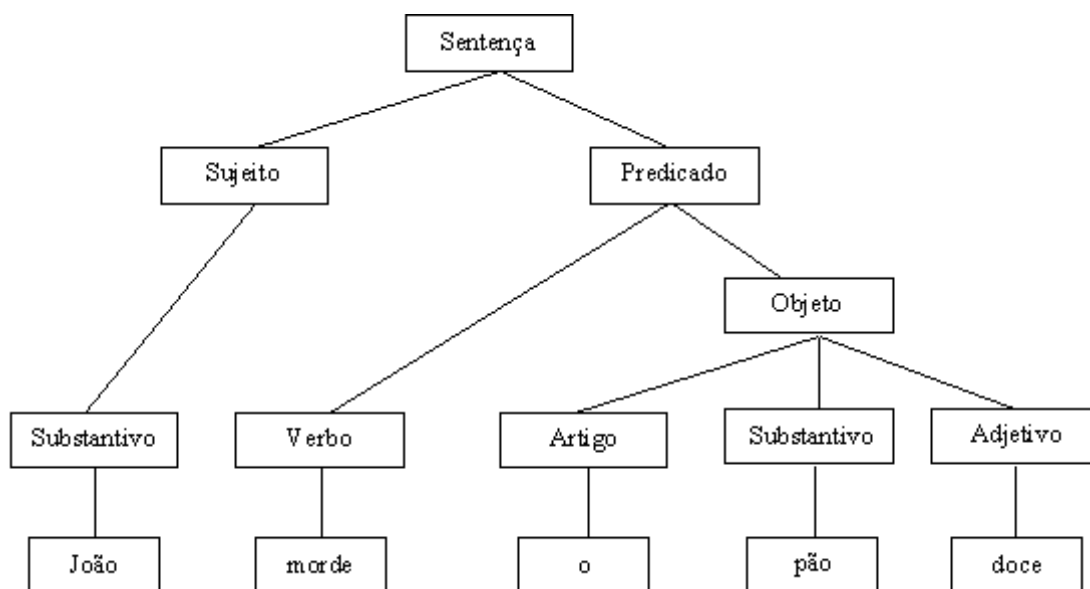


Figura 2-6 - Grafo de uma sentença da língua portuguesa

Se $G = (V, T, P, S)$ é uma gramática, então o conjunto de todas as sentenças que podem ser geradas a partir do símbolo inicial desta gramática, através de derivações sucessivas é chamada de **linguagem gerada** (Hopcroft & Hullman, 1979)².

$$L(G) = \{w \in T^* / S \Rightarrow^* w\}$$

Se $w \in L(G)$, então a seqüência:

$$S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \dots \Rightarrow w_n \Rightarrow w$$

² De forma equivalente, ao caso de conjuntos, o termo \Rightarrow^* denota a aplicação sucessiva do operador \Rightarrow

é uma derivação da sentença (ou palavra) w . As cadeias S, w_1, w_2, \dots, w_n que contêm variáveis, assim como objetos terminais, são chamados **formas sentenciais** da derivação.

Exemplo: Considere a gramática $G = (\{S\}, \{a,b\}, P, S)$ onde P é dado por

$$S \rightarrow aSb$$

$$S \rightarrow \varepsilon$$

Então $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$. Logo podemos escrever $S \Rightarrow^* aabb$.

A cadeia $aabb$ é uma sentença na linguagem gerada por G , enquanto $aaSbb$ é uma forma sentencial.

Uma gramática define completamente $L(G)$, porém pode não ser fácil obter uma descrição explícita da linguagem a partir da gramática. Mas, no exemplo dado, é fácil supor e provar que

$$L(G) = \{a^n b^n / n \geq 0\}$$

Se observarmos que a regra $S \rightarrow aSb$ é recursiva, teremos uma prova por indução.

Todas as formas sentenciais têm a forma $w^i = a^i S b^i$.

Se $i = 0$, $w_0 = S$

Supondo que $w_i = a^i S b^i$ mostramos que $w_{i+1} = a^{i+1} S b^{i+1}$. Aplicando a regra $S \rightarrow aSb$, obtemos $a^{i+1} S b^{i+1}$. Logo, toda forma sentencial tem essa forma.

Agora, aplicando a regra $S \rightarrow \varepsilon$ temos que todas as sentenças têm a forma $a^n b^n$ para $n \geq 0$

Em geral não é fácil achar a gramática para uma linguagem descrita informalmente ou dar caracterização intuitiva da linguagem definida pela gramática. Para mostrar que uma dada linguagem é de fato gerada por uma gramática G devemos mostrar:

Que todo $w \in L$ pode ser derivado de S usando G.

Toda cadeia derivada, usando G, está em L.

Exemplo: Considere a gramática $G_1 = (\{A, S\}, \{a, b\}, P_1, S)$ onde P_1 é dado por

$$S \rightarrow aAb \mid \varepsilon$$

$$A \rightarrow aAb \mid \varepsilon$$

Essa gramática é equivalente (no sentido que gera a mesma linguagem) à gramática do exemplo anterior, ou seja, $L(G) = L(G_1)$. A equivalência é fácil de mostrar provando que $L(G_1) = \{a^n b^n \mid n \geq 0\}$.

2.4.2 Gramática Formal Fuzzy

No presente trabalho iremos substituir o conceito de cadeia, da gramática formal clássica pelo conceito de estruturas $E = \{C, R, M\}$, onde

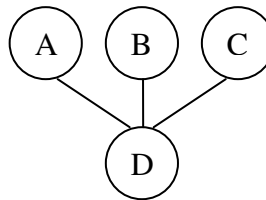
- C é o conjunto de entes abstratos, chamados aqui de partes ou constituintes c_i , que compõe os objetos a serem descritos pela linguagem, isto é $C = \{c_i\}_{i=0}^{\infty}$. O elemento c_0 é chamado aqui de constituinte indeterminado. A noção de constituinte indeterminado não deve ser confundida com a noção de elemento vazio. O elemento vazio não existe e não entra na definição de estrutura porque a aceitação de sua existência resultaria na sua inclusão em toda e qualquer estrutura E . O constituinte indeterminado permite a existência de ignorância sobre a organização de uma estrutura E .

- R é um conjunto de relações entre essas partes, isto é: $R \subset C \times C$

- M representa o estado atual da estrutura

As estruturas podem ser:

- a) Completas: quando todos seus elementos c_i são distintos de c_0 . No exemplo abaixo

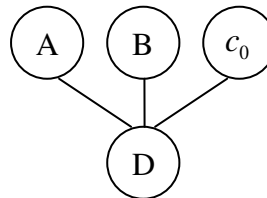


$C = A, B, C, D$

$R = \{(A,D), (B,D), (C,D)\}$

$M = \langle \text{desenho da estrutura} \rangle$

b) Incompletas: quando pelo menos um de seus elementos é c_0



$C = A, B, D, c_0$

$R = \{(A,D), (B,D), (c_0,D)\}$

$M = \langle \text{desenho da estrutura} \rangle$

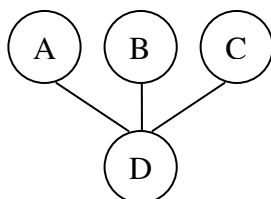
As estruturas podem sofrer transformações por:

- i. Substituição de seus constituintes
- ii. Eliminação de seus constituintes
- iii. Adição de constituintes

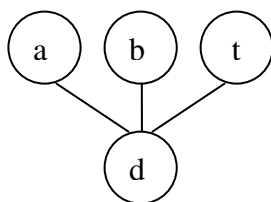
Nem todos $c_i \in C$ é substituível, eliminável ou adicionável. Os constituintes que não podem sofrer as transformações acima em uma estrutura E são chamados de constituintes terminais. Denotaremos aqui esses elementos com os caracteres minúsculos de a a z . Os constituintes que são passíveis dessas transformações em uma estrutura E, são chamados de

constituintes transformáveis e serão denotados com os caracteres maiúsculos de A a Z. O elemento indeterminado c_0 não é transformável como decorrência da ignorância sobre seu estado e será denotado, aqui, por @ .

A estrutura



é uma estrutura totalmente transformável, enquanto que a estrutura



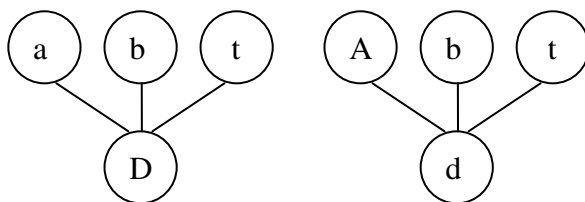
$C = a, b, t, d$

$R = \{(a,d), (b,d), (t,d)\}$

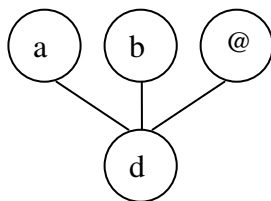
$M = \langle \text{desenho da estrutura} \rangle$

é uma estrutura totalmente terminal.

As estruturas abaixo são estruturas parcialmente transformáveis

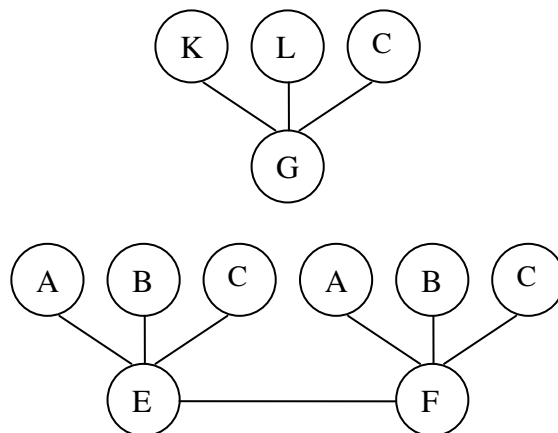


A estrutura abaixo é terminal

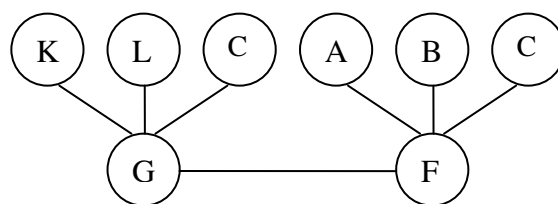


pois o elemento @ não é transformável.

Numa teoria com pouca restrição, dada as estruturas abaixo:



Se $\mu(E, G) > 0$ e $\mu(F, G) > 0$ a substituição do elemento G por E dependeria das similitudes $\mu(A, K)$, $\mu(B, K)$, $\mu(A, L)$, $\mu(B, L)$. Se pelo menos $\mu(A, K) > 0$, $\mu(A, L) > 0$ ou $\mu(B, K) > 0$, $\mu(B, L) > 0$ então a transformação



seria permitida.

Numa teoria mais restrita a transformação só seria permitida se

$$\mu(A, K) = 1, \mu(A, L) = 1, \mu(B, K) = 1, \mu(B, L) = 1.$$

A definição de uma Linguagem Formal e uma gramática \mathbf{G} , descrita acima (Chomsky, 1965, Hopcroft & Ullman, 1979, Simpser, 1997), foi ampliada em (Rocha 1992, Rocha 1997) para que pudesse representar também o conceito de conjuntos nebulosos.

A teoria dos conjuntos nebulosos destaca o conceito da incerteza, que foi incorporado às Linguagens Formais, gerando desta forma, as Linguagens Formais Fuzzy ou Nebulosas.

Uma das principais mudanças foi a inclusão do conceito de incerteza na reescrita (Mizumoto et al., 1973, Negoita e Ralescu, 1975), comparação e avaliação (Rocha, 1992, Rocha, 1997).

Dessa forma, pode-se dizer que uma gramática formal nebulosa (GFN) \mathbf{G} é uma estrutura definida como (Rocha, 1992; Rocha, 1997):

$$\mathbf{G} = (\mathbf{V}, \mathbf{T}, \mathbf{P}, \mathbf{S}, \mu)$$

onde:

- a) \mathbf{V} : é um conjunto finito de símbolos não-terminais;
- b) \mathbf{T} : é um conjunto finito de símbolos terminais;
- c) \mathbf{P} : é um conjunto finito de regras de produção, onde cada produção se encontra na seguinte forma $r: \alpha \rightarrow \beta$, onde α é um símbolo não-terminal e β é uma estrutura de símbolos em $(\mathbf{V} \cup \mathbf{T})^*$.
- d) \mathbf{S} : conjunto inicial não-vazio de palavras $\in \mathbf{V}^+$;
- e) μ : é uma função de $\mu: \Sigma^* \times \Sigma^* \rightarrow [0,1]$, que representa o grau de aplicação de uma produção, ou seja, o grau de similaridade entre duas sentenças.

Definições:

Para representar um alfabeto podem-se utilizar as definições empregadas na gramática formal clássica, ou seja:

$$\Sigma = V \cup T$$

Σ^* = conjunto de todas as sentenças compostas de símbolos de Σ incluindo a sentença vazia.

$$\Sigma^+ = \Sigma^* - \{\varepsilon\}$$

O *grau de similaridade* (comparação) $\mu(s_i, s_j)$ de duas estruturas s_i, s_j é uma medida do espaço cartesiano ($\Sigma^* \times \Sigma^*$) para o intervalo fechado $[0, 1]$. Em outras palavras:

$$\mu : \Sigma^* \times \Sigma^* \rightarrow [0, 1]$$

$$\mu(s_i, s_j) \rightarrow 1 \text{ se } s_i \text{ tende a ser igual a } s_j;$$

$$\mu(s_i, s_j) \rightarrow 0 \text{ caso contrário.}$$

Ou seja, o grau de similaridade entre uma palavra e ela mesma deve ser sempre 1.

Seja $G = (V, T, P, S, \mu)$ uma GFN. Uma *Derivação* é um par da relação denotada por \Rightarrow com domínio em $(V \cup T)^+$ e contra-domínio em $(V \cup T)^*$. Um par (α_1, α_m) é representado de forma infixada, como segue:

$$\alpha_1 \Rightarrow^* \alpha_m$$

A seqüência de derivação $\alpha_1 \Rightarrow^* \alpha_m$ da estrutura $\alpha_m \in \Sigma^*$ de G é o conjunto ordenado de reescritas requeridas para transformar o símbolo inicial $\alpha_1 \in S$ em α_m . Em outras palavras:

$$\alpha_1 \Rightarrow^* \alpha_m \text{ se}$$

$$\alpha_1 \Rightarrow \alpha_2, \alpha_2 \Rightarrow \alpha_3, \dots, \alpha_{m-1} \Rightarrow \alpha_m,$$

$$\alpha_1, \alpha_2, \dots, \alpha_m \in (V \cup T)^*$$

O processamento de qualquer sentença da seqüência de derivação $s_i \Rightarrow^* s_m$ é um conjunto **seqüencialmente ordenado** de operações de reescrita, cada uma envolvendo os seguintes passos:

1) **Comparação:** o lado esquerdo s_i da provável regra de reescrita é comparado com os símbolos da estrutura s_j a ser processada. Se essa comparação for bem sucedida, ou seja, $\mu(s_i, s_j) > \theta$, então:

2) **Reescrita:** a subestrutura selecionada em s_i é substituída pelo lado direito s_j da regra de reescrita aceita. Finalmente:

3) **Avaliação:** a pertinência da estrutura derivada s_j a $L(G)$ é calculada como uma medida definida no intervalo fechado **[0,1]**.

O **grau de avaliação** de s_i como uma **fbf** de L é calculado através do grau de similaridade $\mu(s_i, s_j)$ de s_i para com a estrutura $s_j \in L(G)$. Sendo assim:

$$\mu(s_i, s_j) \rightarrow [0,1]$$

$$\mu(s_i, s_j) \rightarrow 1 \text{ se } s_i \text{ tende a ser uma fbf, já que tende a ser igual a um elemento de } L(G);$$

$$\mu(s_i, s_j) \rightarrow 0 \text{ caso contrário.}$$

A Gramática Nebulosa

$G = (V, T, P, S, \mu)$, está associada a uma linguagem nebulosa $L(G)$ (Mizumoto, 1973; Negoita & Ralescu, 1975; Rocha, 1997; Rocha et al., 1980).

$$L(G) = \{w \in T^* \mid \alpha \Rightarrow^+ w \text{ e } \alpha \in S\}$$

Caso Especial – Linguagem Formal Clássica

Se:

a -) o grau de similaridade μ ser definido como:

$$\mu : \Sigma^* \times \Sigma^* \rightarrow [0, 1]$$

$$\mu(s_i, s_j) = 1 \text{ para } s_i = s_j$$

0 caso contrário

b -) o grau de avaliação σ ser definido como:

$$\sigma(s_j, T^*) = 1 \text{ se } s_j \in \square T^*$$

0 caso contrário

Então temos uma linguagem formal clássica conforme discutida por Chomsky e outros (Chomsky, 1965; Searls, 1993).

2.4.3 Teoria Computacional das Percepções

Os seres humanos possuem uma notável capacidade de realizar uma grande variedade de tarefas físicas e mentais sem a necessidade de medições ou cálculos precisos. Os valores dos atributos são granulados e cada grânulo reúne um grupo de valores (pontos, objetos) por similaridade, proximidade ou função. Por exemplo, os grânulos de uma distância podem ser marcados como muito perto, perto, médio, longe, muito longe, dentre outros. Quando esses grânulos têm limites imprecisos, são chamados Grânulos Fuzzy ou F-grânulos (Zadeh, 2001).

A Teoria Computacional de Percepções (TCP), proposta por Zadeh (2001) como um paradigma do raciocínio impreciso, pressupõe que:

1. As percepções humanas são organizadas a partir de F-grânulos;
2. São muitas vezes descritas em linguagem natural, como por exemplo, o objeto X está longe (perto...) e aproximadamente à direita (esquerda...) do objeto Y.

Na TCP, uma proposição p é vista como uma resposta a uma questão q e o significado de p é representado como uma restrição generalizada R . O cálculo com percepções é proposto como um processamento de uma Linguagem de Restrições Generalizadas (LRG), de modo que a propagação das restrições tem como meta responder uma determinada questão (Zadeh, 2001).

Por exemplo, a busca de uma determinada imagem põe solicitação q do tipo:

“Localizar o objeto X a uma relação espacial de Y”

que requer como solução dois tipos de proposições:

- p_1 : sobre as relações de distância entre os objetos na imagem; e
- p_2 : sobre as direções entre esses mesmos objetos.

Em outras palavras, o reconhecimento visual de uma imagem c_k pode ser feito respondendo a uma questão q referente à localização de objetos que compõe c_k :

$$(I) \quad q = (p_1, p_2) = \text{Localização}(O_x, O_y)$$

E a questão q poderia ter duas respostas:

$$(II) \quad p_1: \text{A distância entre } (O_x, O_y) \text{ é } \{\text{Muito perto, Perto, Média, Longe, Muito Longe}\}$$

$$p_2: \text{A direção entre } (O_x, O_y) \text{ é } \{\text{Norte, Sul, Leste, Oeste, Noroeste...}\}$$

De modo que a solução é encontrada com o tratamento das restrições:

$$(I) \quad R_1 = \{\text{Muito Perto, Perto, Média, Longe, Muito Longe}\}$$

$$R_2 = \{\text{Norte, Sul, Leste, Oeste, Noroeste...}\}$$

A análise completa de c_k , realizada como cálculo da localização do conjunto de objetos O_i em relação a uma referência Z , poderia ser formalizada do seguinte modo:

$$(II) \quad c_k = [q_1, q_2, \dots, q_n], \text{ onde:}$$

$$q_i = (p_1, p_2) = \text{Localização}(O_i, Z)$$

o que significa que a questão q é repetida para todos os objetos presentes na cena c_k . A referência Z pode ser o centro de um objeto específico O_r , ou um ponto qualquer no espaço XY .

2.4.4 Gramática e Percepção Computacional do Sistema

O sistema de reconhecimento de imagens, desenvolvido no presente projeto, está baseado em uma gramática formal nebulosa

$$G = \{V, T, P, S, \mu\}$$

onde;

V – São os grânulos fuzzy (F-grânulos) presentes em uma imagem.

T – Imagem a ser analisada.

P – Regras, definidas pelo usuário, contendo as informações que identificam o objeto a ser pesquisado.

S – São as imagens ou segmentos da imagem que satisfaçam as regras de P.

μ - Grau de pertinência de semelhança entre a imagem S e o grafo T definido por P.

Para a obtenção dos V's de cada imagem T, é necessária a aplicação de técnicas de processamento digital de imagens, tais como:

- Cálculo de histograma;
- Conversão de cores;
- Extração de contornos;
- Rotulação, manipulação e comparação de áreas;
- Extração de pontos significativos; dentre outros.

Podemos dizer que os F-grânulos V são os descritores de uma imagem, ou seja, o conjunto de características visuais, tais como a cor, relações espaciais e formas,

$$V = \{ v_i \}, i = 1..k.$$

Para obtenção de T, temos que fazer uma busca em todos os elementos S de uma determinada base de dados, analisando o grau μ de semelhança de S em relação a P, a partir de V.

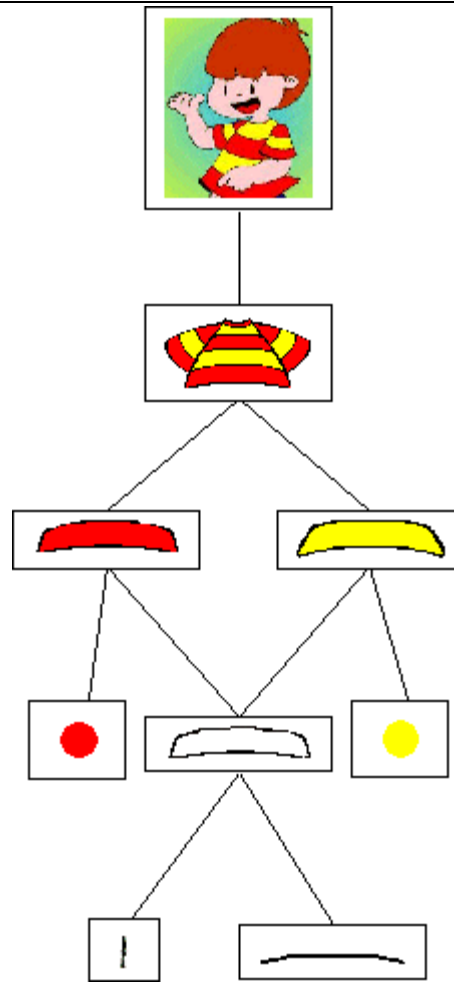


Figura 2-7 - Grafo da personagem Juca

Para o caso do objeto ilustrado na Figura 2-7, teríamos a gramática

$G_{juca} = \{V_{juca}, T_{juca}, P_{juca}, S_{juca}, \mu\}$ onde,

$V_{juca} = \{<Cor>, <Áreas>, <Posição>, <Distância>, <Relações>, <Segmento de reta>\}$

$T_{juca} = \{ (r, g, b) \mid 0 \leq r \leq 255, 0 \leq g \leq 255, 0 \leq b \leq 255 \}$

$P_{juca} = \{Juca \rightarrow <Camisa do Juca >\}$

$<Camisa do Juca > \rightarrow <Listra> \text{ Vermelha "Muito perto"} \text{ de } <Listra> \text{ Amarela "Muito perto"} \text{ de } <Listra> \text{ Vermelha}$

<Listra> → <Área Retangular><Cor>,

<Área Retangular> → Segmento de reta Horizontal, Segmento de reta Vertical, Segmento de reta Horizontal, Segmento de reta Vertical

<Cor> → Vermelha | Amarela | Azul

}

$S_{juca} = \langle \text{Juca} \rangle$

$\mu = \{ \text{Grau de pertinência associado às variáveis lingüísticas presentes em P} \}$.

A Figura 2-8 representa, na forma de grafo, o objeto S_{juca} , reconhecido a partir de

$P_{juca} = \{ \langle \text{Camisa do Juca} \rangle \}$.

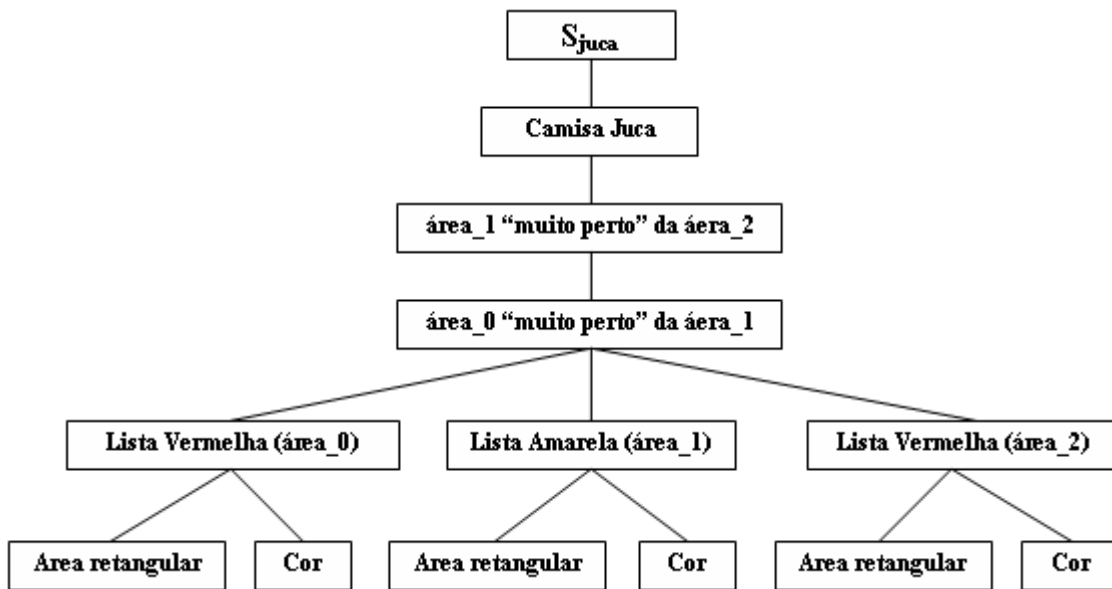


Figura 2-8 -Grafo S_{juca}

Para chegar a um resultado de S_{juca} , ou seja, o conjunto de imagens que contenham objetos que satisfaçam a uma regra ou ao conjunto de regras P_{juca} , é necessário percorrer todas as imagens T_{juca} presentes na base de dados analisando o grau de semelhança μ à regra P_{juca} a partir de V_{juca} .

De forma semelhante pode ser feita uma análise para a personagem Laura, conforme a Figura 2-9.

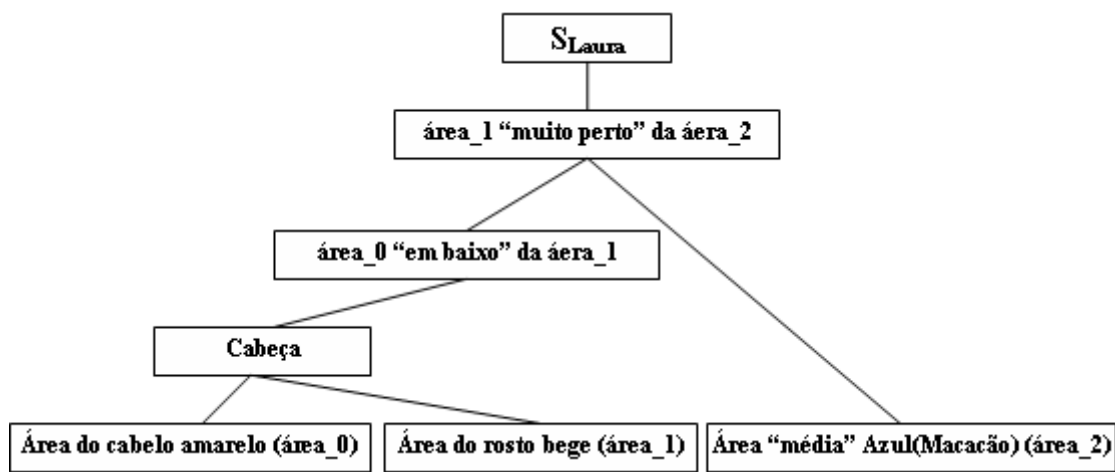


Figura 2-9 - Grafo do objeto Laura

Para encontrar imagens em que ocorra a existência de uma cena, como por exemplo, o objeto Juca perto e ao lado direito do objeto Laura, basta criar as regras P em relação aos objetos já montados anteriormente e não aos descritores presentes em cada imagem. Com isso, o objeto todo, e não só seus descritores, começa a fazer parte de V na nova consulta.

Na Figura 2-10 é mostrado o grafo do exemplo citado acima.

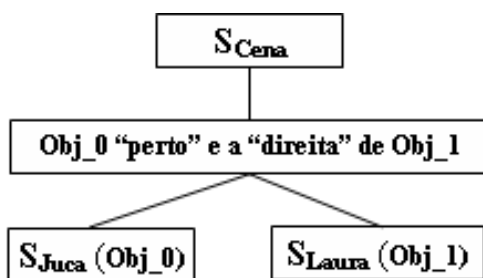


Figura 2-10 - Grafo de uma cena

3 O SISTEMA SENSOR v2

3.1 Primeiras versões

Em Serapião(1996) foi desenvolvido um sistema sensorial computacional distribuído, denominado SENSOR (v0), voltado ao reconhecimento de padrões, através da abordagem sintática. A ideia da abordagem sintática é que dado certo padrão, ele possa ser dividido em partes menores, cuja representação possa ser feita através de uma estrutura hierárquica, como, por exemplo, uma estrutura tipo árvore. A teoria de linguagens formais nebulosas pode ser utilizada para simular o funcionamento do sistema visual humano e o sistema **Sensor** tem a capacidade de utilizar linguagens formais definidas por gramáticas para recodificar informações numéricas contidas em uma imagem.

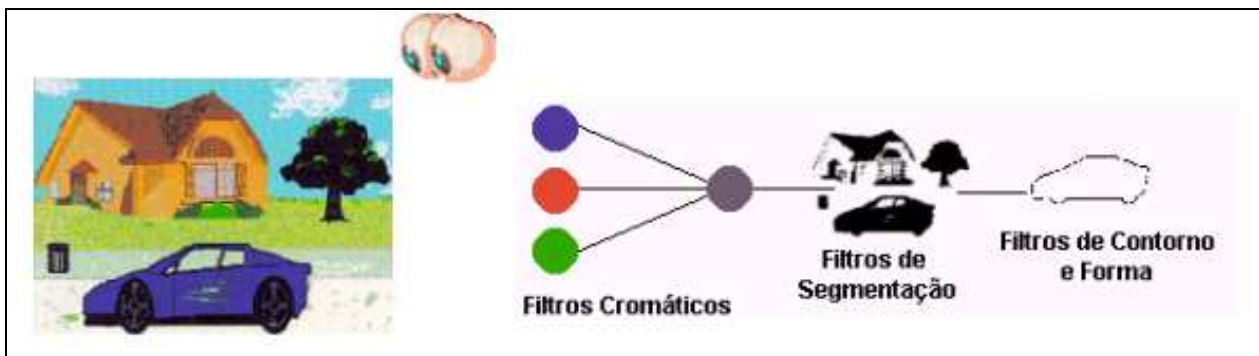


Figura 3-1 - Organização computacional dos filtros

O processamento no nível da retina pode ser simulado por vários agentes responsáveis pela aplicação de diversos tipos de filtros e algoritmos para extração de contorno, enquanto que agentes responsáveis pelos filtros de segmentação podem simular circuitos pré-corticais.

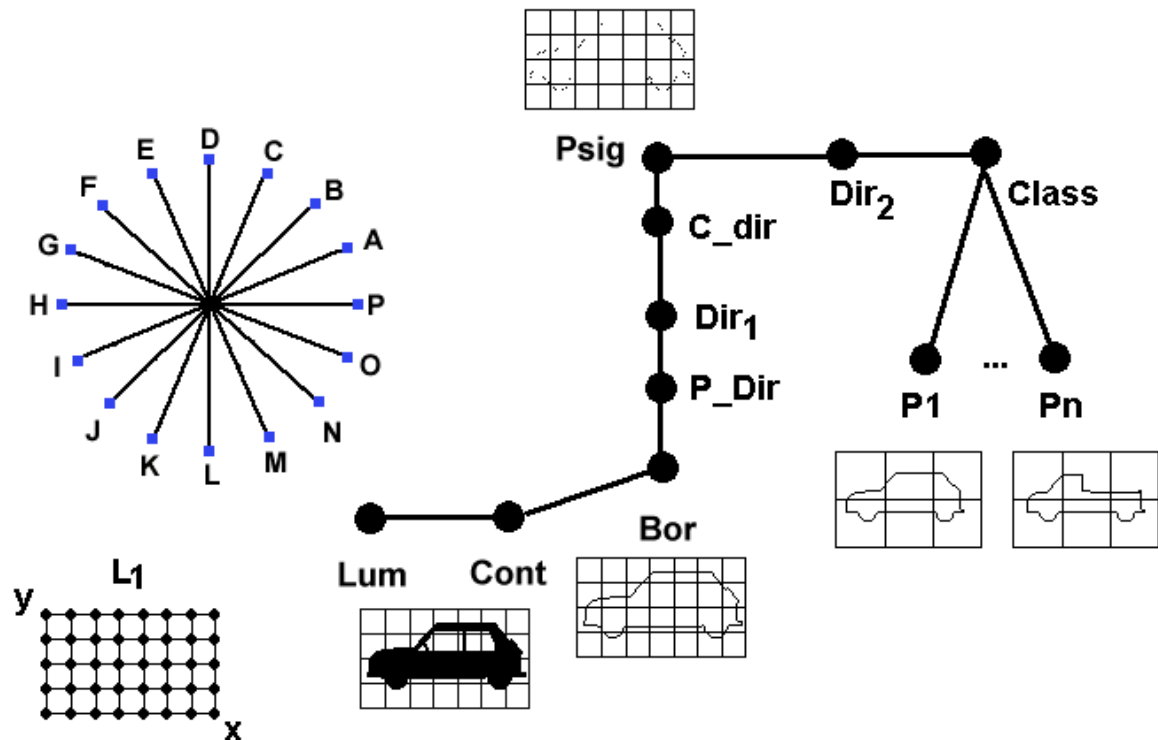


Figura 3-2 - Estrutura de agentes utilizados na simulação do sistema visual humano.

A partir da medida de luminância medida pelo agente **Lum**, o agente primário **Cont** recodifica a imagem gerada em uma sentença de caracteres pertencentes a um dicionário

C_1 {p = preto, b = branco}

e suportada por uma gramática regular.

O agente **Bor** na Figura 3-2, reescreve a sentença em outra sentença suportada por uma gramática regular definida sobre o seguinte dicionário $C_2 = \{n, f\}$ de acordo com as regras: pb \rightarrow n, bp \rightarrow n, bb \rightarrow f, pp \rightarrow f, assim descrevendo a borda do objeto sensoriado.

Com as informações passadas por **Bor**, o agente **P_Dir** gera uma sentença s_3 que descreve as direções entre dois caracteres consecutivos da Borda, utilizando o dicionário C_3 , onde: $C_3 = \{p=\text{positivo}, n=\text{negativo}, z=\text{zero}, a=\text{vazio}, b=\text{pequeno}, c=\text{médio}, d=\text{grande}, e=\text{enorme}\}$ e $S_3 = \{c_{x1} c_{y1} c_2\}^j$ tal que $c_{x1}, c_{y1} \in \{p, n, z\}$, $c_2 \in \{b, c, d, e\}$

O agente **Dir₁** reescreve **s₃** em outra cadeia **s₄** suportada pela gramática **G₄** utilizando o seguinte dicionário: $C_4 = \{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P\}$,

de acordo com as seguintes regras:

(I) 

onde “*” significa “qualquer direção”.

Por exemplo, a cadeia gerada pelo agente **Dir₁**, que descreve o contorno do carro na Figura 3-2 é dada por:

HDDCCPCPPPPCCCCPPPPPPPPPPMPMCMLLKKHEEEHHHHHHHHHHHHHHHHKKEHEHHHD

Na seqüência, o agente **C_Dir** reescreve a cadeia **s₄** em uma cadeia **s₅** composta por símbolos desse dicionário: $C_5 = \{z = \text{Sem Mudança}, c = \text{Mudança no Sentido Anti-horário}, d = \text{Mudança no Sentido Horário}\}$, descrevendo as mudanças de direções entre os pontos da figura, associadas com dois símbolos consecutivos de **s₄**. Assim, **s₅** descreve as mudanças de direção da borda da figura do carro.

O agente **Psig** pesquisa a existência das subcadeias: **zcc, dcc, zdd, cdd, czc, dzd, zzc, zzd, zcz, e zdz**, em **s₅** para gerar um vetor **v₆** contendo coordenadas **xy** da localização das subcadeias. Este vetor descreve os pontos da borda **Psig** (pontos significativos) onde as mudanças de direções ocorreram.

O agente **Psig** usa a seguinte gramática **G₅** sobre o dicionário **C₅**, para gerar uma cadeia **s₆** de acordo com as seguintes regras:

(II)
$$\left\{ \begin{array}{llll} zcc \rightarrow S & dcc \rightarrow S & zdd \rightarrow S & cdd \rightarrow S \\ czc \rightarrow S & dzd \rightarrow S & zzc \rightarrow S & zzd \rightarrow S \\ zcz \rightarrow S & zdz \rightarrow S & & \end{array} \right\}$$

Senão $\rightarrow N$,

produzindo uma cadeia do tipo:

(III) $S_6 = \{ \text{SNN.....} \}$

Agora, o agente **Dir₂** (Figura 3-2) recodifica s_6 em uma cadeia s_7 descrevendo as principais direções dentre os pontos **Psig** da borda **B₂**. No exemplo abaixo, somente os pontos significativos do objeto carro da Figura 3-2

(IV) $S_7 = \{ \text{HDDBBPCCPPMMCMLKKFEHHJKFFHD} \}$

A partir da informação gerada por s_7 , podemos agora armazenar essa informação como sendo um modelo de características de um objeto carro que é inserido em uma base de dados de conhecimento.

Com a definição anteriormente descrita, novas imagens podem ser classificadas e seus objetos podem ser confrontados com a base de dados de conhecimento, a fim de retirar informações necessárias para a identificação de objetos.

Já Rodella (2002) desenvolveu um sistema (SENSOR v1) para análise de um conjunto de cenas composta por elementos pertencentes a um conjunto definidos de objetos (Figura 3-3), para com isso identificar as associações mais frequentes entre esses objetos (Figura 3-4).



Figura 3-3 - Objetos para descrever uma cena



Figura 3-4 - Cena Prototípica Média

3.2 Versão Atual

O sistema SENSOR – v2 é um sistema de recuperação visual de imagens, baseado no sistema visual natural, que utilizou e ampliou as técnicas de reconhecimento de padrões implementadas no SENSOR v0 e as técnicas de descrição de objetos e cenas utilizadas no SENSOR v1.

Este capítulo visa dar uma visão geral do funcionamento do sistema. Serão descrito os passos para inserção das imagens na base de dados, a criação das sentenças dos objetos a serem procurados bem como a busca da sentença na base de dados.

3.2.1 Inserção e compilação das imagens da base de dados

A primeira etapa para a utilização do sistema consiste na inserção e compilação da base de dados de figuras, para gerar as descrições das imagens de acordo com os filtros escolhidos ou calibrados pelo usuário.

Ao inserir uma imagem na base de dados são armazenadas as informações de caminho de origem do arquivo e data e hora da última modificação. Essas informações serão utilizadas para carregar a imagem no sistema e analisar a versão da imagem para o sistema não trabalhar com dados desatualizados.

Na primeira vez em que o sistema utilizar a imagem para processamento de busca, essa é compilada e as informações extraídas desse processamento são armazenadas na base de dados. Essa compilação consiste em encontrar todas as áreas presentes na imagem e convertê-las em uma gramática. Os passos realizados pelo sistema são:

- Leitura da imagem;
- Aplicação de filtros;
- Rotulação;
- Extração das áreas da imagem; e
- Armazenamento destas áreas na base de dados.

A Figura 3-6 mostra o resultado da aplicação de alguns filtros disponíveis em uma imagem inserida (Figura 3-5) no sistema e a Figura 3-7 mostra uma área desta imagem e sua sentença.



Figura 3-5 – Imagem Inserida no sistema



Figura 3-6 – Resultado do processamento. Áreas e rotulação, cada cor representando uma área.

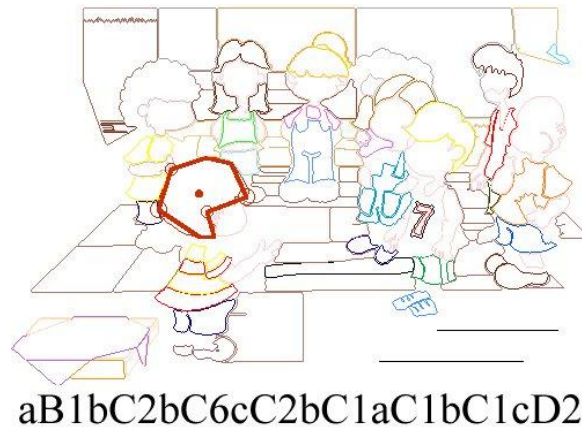
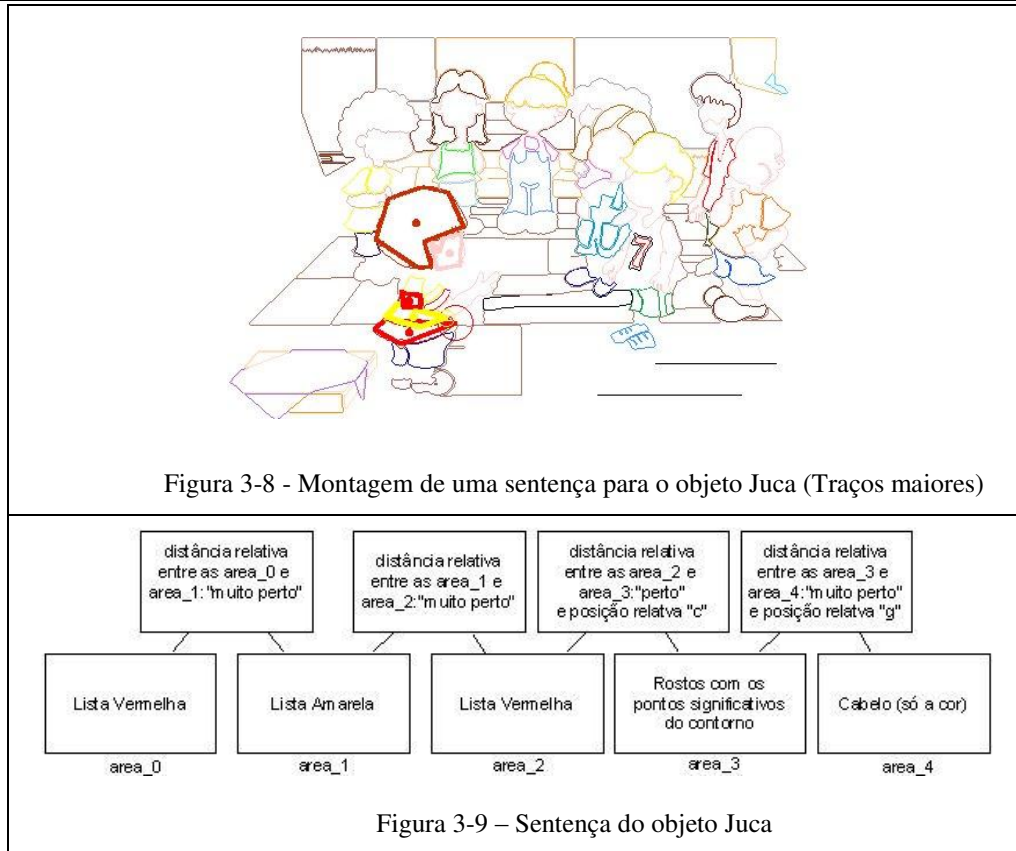


Figura 3-7 – Imagem somente com os contornos e em destaque os pontos significativos da cabeça do personagem Juca e a respectiva sentença desta área

Os processos de compilação e conversão da imagem em áreas serão detalhados no capítulo 4.

3.2.2 Construindo uma sentença

Para criar a sentença de um objeto, que será o modelo de busca para o sistema, deve-se escolher uma única figura na base de dados que contenha esse objeto e criar as relações das partes que o compõem. Essas relações podem ser as mais variadas possíveis, com preferências às particularidades de cada objeto. Por exemplo, o personagem Juca da base de dados do Projeto ENSCER, o qual foi utilizado para o presente trabalho, tem como característica marcante a camisa com listras amarelas e vermelhas. Para montar uma, das várias sentenças possíveis, desse personagem, foram selecionados da imagem uma listra vermelha, uma listra amarela e outra listra vermelha com uma distância “muito perto”. Também foi inserido o contorno (seus pontos significativos) do rosto com uma distância “perto” e a posição relativa “c” (em cima) entre as listras da camisa e, por último, foi inserida a cor do cabelo com a relação de distância “muito perto” na posição “g” (em baixo) do rosto. Na Figura 3-8 temos uma visão gráfica da sentença do objeto Juca e na Figura 3-9 suas informações e relações.



3.2.3 O processamento das buscas e os resultados encontrados

Para realizar uma busca, basta escolher uma sentença e o sistema percorrerá todas as imagens da base de dados fazendo comparações e calculando a pertinência da existência do mesmo em cada imagem. As imagens que estiverem acima do limiar definido pelo usuário (0.5 como padrão) são consideradas imagens que contêm a sentença desejada, e abaixo desse limiar, imagens que não as contêm. Na Figura 3-10, temos o exemplo de um dos resultados da busca retornada pelo sistema. Já na Figura 3-11, temos as áreas que o sistema levou em conta para a tomada da decisão.

Ao final da comparação é exibida uma lista com todas as imagens candidatas, bem como o grau de certeza de que possuem a sentença desejada.

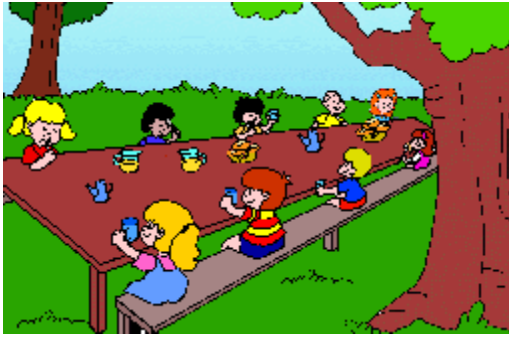


Figura 3-10 – Imagem em que o sistema reconheceu o objeto Juca



Figura 3-11 - Objeto reconhecido na Imagem

4 Estrutura do sistema SENSOR v2

Este capítulo visa descrever a estrutura básica do sistema de busca gráfica implementado no presente trabalho.

4.1 Estrutura do sistema

O sistema está estruturado em dois módulos principais. O primeiro módulo, desenvolvido em Visual C, é o controle activeX do sistema (OCX), responsável pelo processamento de cada imagem. Os principais métodos do processamento disponível no OCX são:

- Cálculo do histograma de imagens coloridas e em nível de cinza;
- Cálculo do histograma acumulado;
- Equalização de histograma;
- Conversão da imagem colorida em níveis de cinza;
- Binarização da imagem;
- Extração dos contornos da imagem;
- Métodos para manipulação e comparação de áreas;
- Extração de pontos significativos.

O segundo módulo do sistema é a interface com o usuário que foi desenvolvido em Visual Basic por oferecer uma integração com o Visual C e disponibilizar controles úteis para o desenvolvimento do projeto, dentre eles podemos destacar o PictureBox, um controle usado para exibir imagens com várias funcionalidades gráficas incorporadas. Outro controle importante oferecido é o ScriptControl, que possibilitou incorporar uma linguagem de script dentro do sistema. Essa linguagem de script permite que sejam incorporados e processados códigos (escritos em VBScript ou JavaScript) em tempo de execução do sistema, utilizado para refinar a

busca das sentenças a serem procuradas. A Figura 4-1 mostra o diagrama com os componentes do sistema e suas relações.

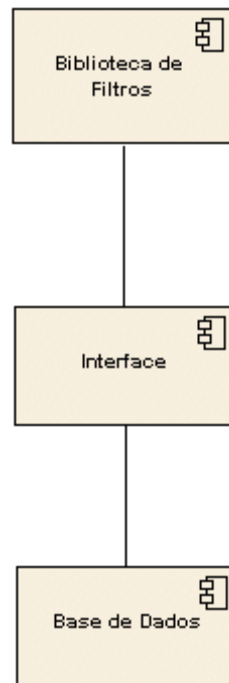


Figura 4-1 - Diagrama de componentes

A modelagem do sistema foi baseada na orientação a objetos descrita em Rumbaugh(1994) e Larman 2000, que trata os objetos de uma maneira abstrata utilizando modelos fundamentados em conceitos do mundo real. A notação padronizada UML foi utilizada nessa modelagem.

4.1.1 Diagramas do sistema

Na modelagem do sistema, foram criados vários diagramas para se ter uma visão geral do sistema. Abaixo serão exibidos esses diagramas.

4.1.1.1 Diagrama de casos de uso

O sistema desenvolvido teve três casos de uso. Esses casos de uso podem ser visualizados na Figura 4-2.

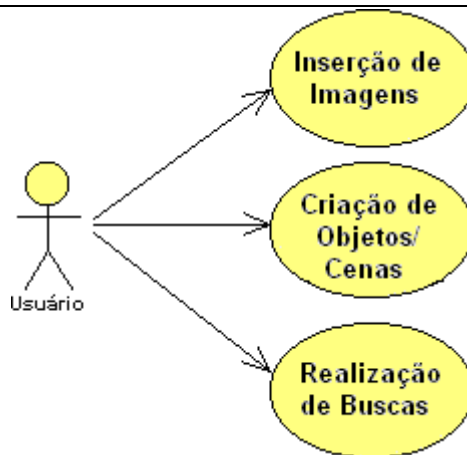


Figura 4-2 - Diagrama de casos de uso

4.1.1.2 Diagrama de atividades

Para cada caso de uso, temos um diagrama de atividades.

Na Figura 4-3, temos o diagrama de atividades para a inserção de imagens no sistema.

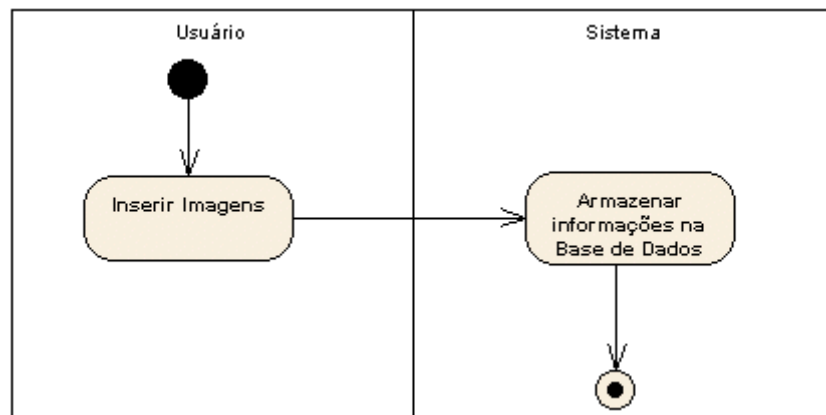


Figura 4-3 - Diagrama de Atividades - Inserindo uma imagem

A Figura 4-4 mostra o diagrama de atividades na criação das sentenças.

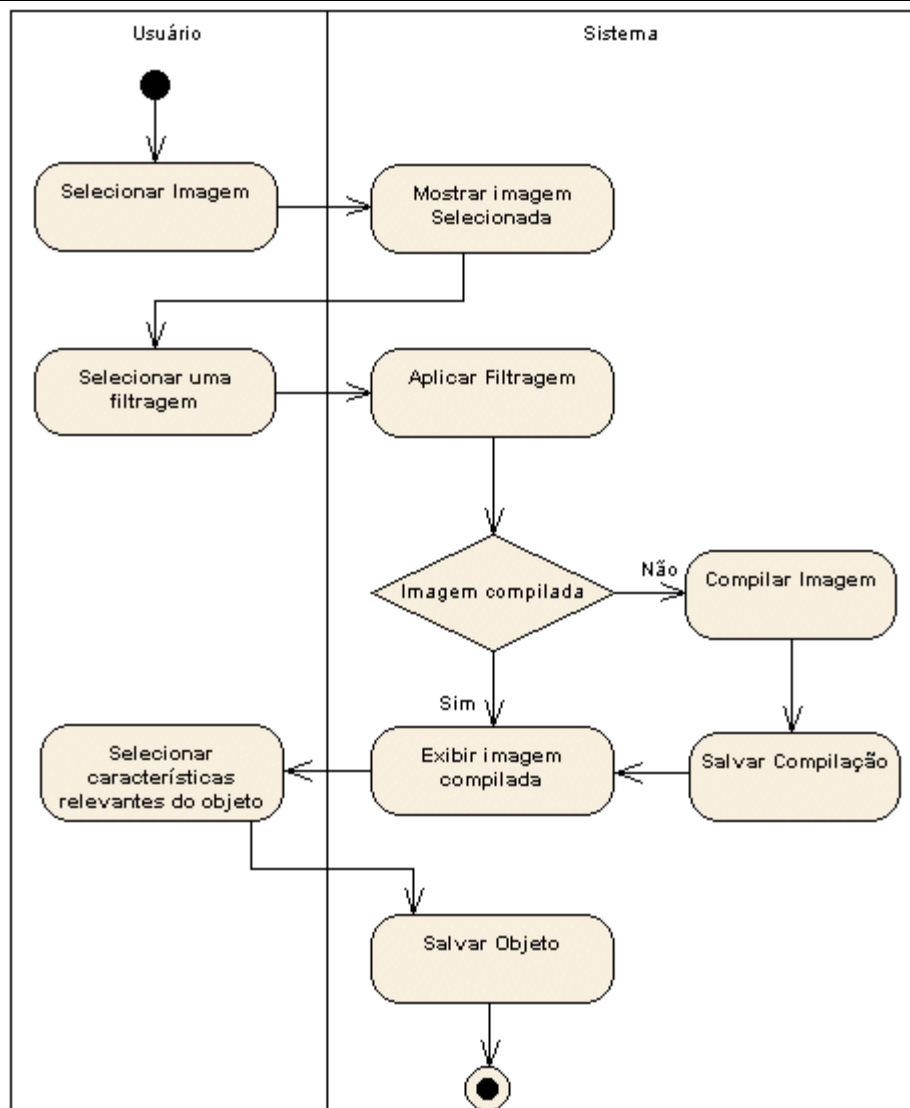


Figura 4-4 - Diagrama de Atividades - Criação de Objetos

O processamento da busca pode ser visualizado no diagrama de atividades mostrado na Figura 4-5.

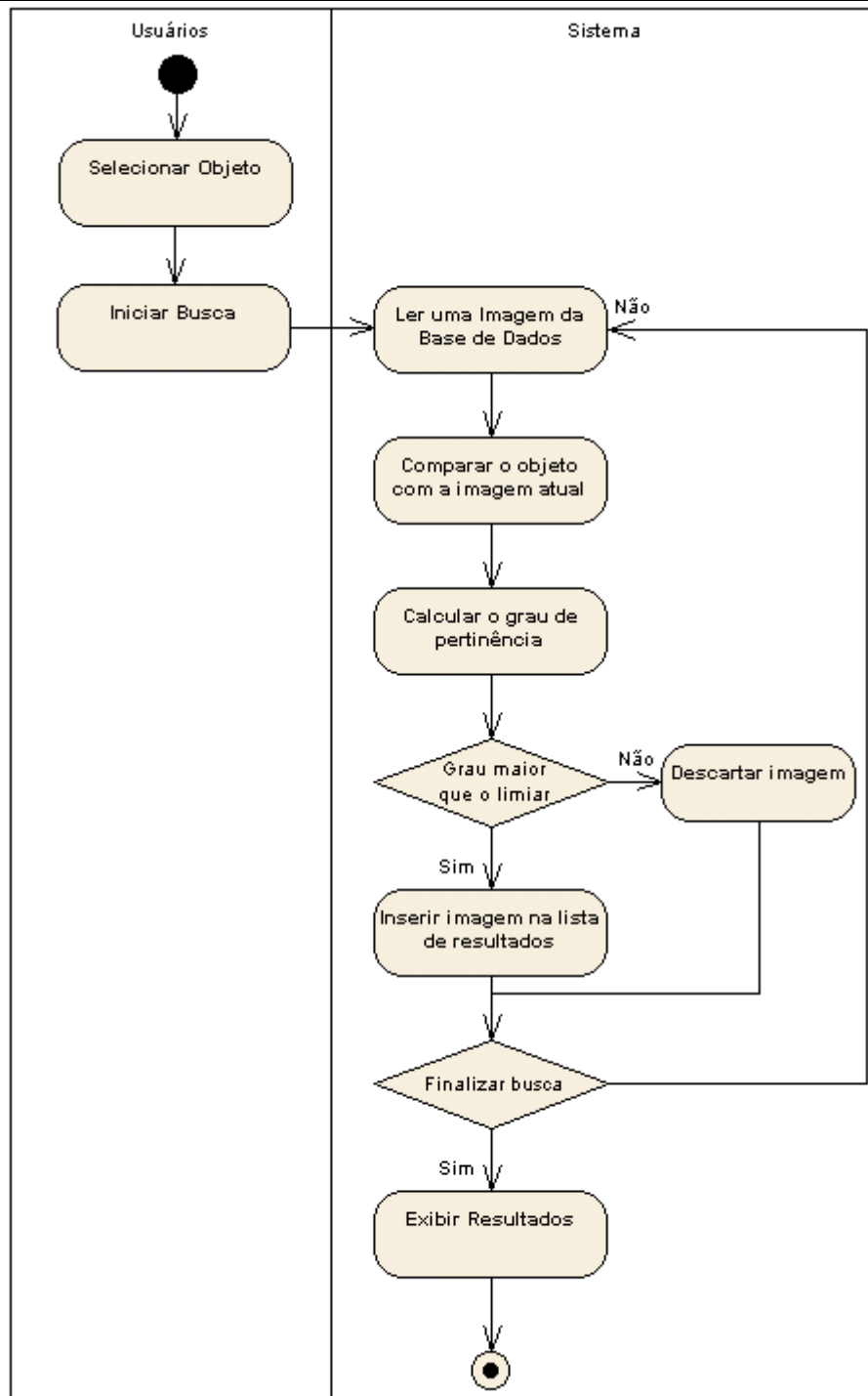


Figura 4-5 - Diagrama de Atividades - Pesquisa das sentenças

4.2 Extração de informações das imagens

As informações são extraídas no momento em que a imagem é inserida no sistema, na montagem da sentença de busca e no seu processamento.

Para permitir uma melhor flexibilidade do sistema, os filtros a serem aplicados nas imagens são programados através de macros, dos quais se pode definir não só os parâmetros dos filtros, como também as ordens com que serão aplicados e quais filtros serão aplicados a cada imagem.

4.2.1 Filtros de Imagem

Os filtros disponíveis no sistema estão implementados no módulo OCX. O componente OCX constitui o núcleo de todo o processamento de imagem do sistema.

Cada instância do OCX contém uma única figura. Para trabalhar com várias figuras deve-se usar várias instâncias do componente, onde se pode copiar a figura atual de um para outro através da função Copy.

A figura é armazenada com o eixo X da esquerda para a direita e o eixo Y de baixo para cima, ambos com origem em 0 (o ponto de baixo à esquerda é o ponto 0,0).

No Apêndice B, são detalhadas todas as funções implementadas para o processamento de imagens e os filtros utilizados durante o desenvolvimento do sistema.

5 Resultados

Neste capítulo, serão mostrados os resultados obtidos com os testes do sistema e algumas comparações com outros sistemas disponíveis na literatura da área.

A principal medida para avaliar um sistema de recuperação de imagem é a “performance” ou desempenho do sistema em relação aos resultados obtidos para as consultas dos usuários e na relação custo/tempo de execução.

Na hora de testar o grau de certeza de um sistema que usa a recuperação de imagens baseadas em similaridade de conteúdo, a validação do sistema é feita através de uma análise manual do conjunto de todas as imagens relevantes a uma dada consulta no banco de dados (*conjunto relevantes*), e comparado com o conjunto resultado que o algoritmo de processamento do sistema automaticamente recupera (*conjunto resultado*).

Para a realização dos testes do sistema, foi utilizada a base de dados imagens do Projeto ENSCER. Essa base possui aproximadamente 22000 figuras. Na fase de implementação do sistema foram utilizadas 970 imagens levando em conta o tamanho e o posicionamento dos personagens nas imagens para que tivéssemos as maiores variações de imagens possíveis. Para os testes finais, visando principalmente analisar desempenho e robustez, o universo de pesquisa contou com 10% do número total de imagens, as quais foram escolhidas aleatoriamente.

Para a criação das sentenças de buscas (consultas) foram utilizadas personagens típicas e objetos mais comuns do material utilizado no projeto Enscer.

Os resultados considerados certos são aqueles que apresentaram um índice de semelhança igual ou superior a 50%. Esse índice pode ser alterado no software antes ou depois do processamento dos resultados.

Para testar o sistema foi escolhido um conjunto de consultas e foram aplicados os seguintes índices (Gong, 1998):

recall (revocação) Do conjunto relevante de uma consulta, o valor do recall é a porcentagem de imagens nesse conjunto que foi encontrada no conjunto resultado, ou seja, o resultado da relação entre o número de informações relevantes recuperadas (**R**) e o total de informações relevantes armazenadas (**A**).

$$IR = \frac{R}{A}$$

Precisão. Do conjunto resultado, o valor de precisão é a porcentagem de imagens nesse conjunto que pertence ao conjunto relevante da consulta, ou seja, o resultado da relação entre o número de informações relevantes recuperadas (**R**) e o total de informações recuperadas (**T**).

$$IP = \frac{R}{T}$$

Quando o valor de revocação for muito superior ao de precisão significa que o sistema retornou um grande número de resultados falsos positivos.

Também será apresentada a sensibilidade, especificidade do valor preditivo positivo (VP+) e do valor preditivo negativo (VP-) para os teste de retroalimentação.

A sensibilidade de uma busca é a probabilidade de que ela classifique corretamente uma imagem que contenha o objeto procurado. A maneira utilizada para calcular a sensibilidade foi pela fração verdadeiro-positivo (FVP), que é a fração do número total de imagens, que contém o objeto, para a qual a busca retorna positiva:

$$FVP = \frac{VP}{VP + FN}$$

onde, VP são os resultados positivos para imagens que de fato possuem o objeto pesquisado e FN são os resultados negativos para imagens que possuem o objeto pesquisado.

A especificidade de uma busca é a probabilidade de que ela classifique corretamente uma imagem que não contenha o objeto procurado, ou seja, a probabilidade de ser negativo para uma imagem que não contenha o objeto. A maneira utilizada para calcular a especificidade foi

pela fração verdadeiro-negativo (FVN), que é a fração do número total de imagens, que contém o objeto, para a qual a busca retorna negativa:

$$FVN = \frac{VN}{VN + FP}$$

onde, VN são os resultados negativos para imagens que de fato não possuem o objeto pesquisado e FP são os resultados positivos para imagens que não possuem o objeto.

Define-se o valor preditivo positivo (VP+) de uma busca como a probabilidade de que uma imagem contenha o objeto, dado que a busca retornou positiva para ela.

$$VP+ = \frac{VP}{VP + FP}$$

Igualmente, define-se o valor preditivo negativo (VP-) de uma busca como a probabilidade de que uma imagem não contenha o objeto, dado que a busca retornou negativa para ela.

$$VP- = \frac{VN}{VN + FN}$$

Retroalimentação é a busca de um objeto realizada a partir dos resultados de buscas anteriores desse mesmo objeto. Neste trabalho, retroalimentação é sinônimo de refinamento.

Os resultados da busca serão mostrados no seguinte formato:

Tabela 5-1 - Tabelas de resultados

Resultados das buscas	Imagens Relevantes	Imagens Não Relevantes	Total
Resultados Positivos	VP	FP	VP+FN
Resultados Negativos	FN	VN	FN + VN
Total	VP + FN	FP +VN	

Objeto Juca

Este é um personagem masculino caracterizado pela bermuda azul, a camisa de listras vermelhas e amarelas e o cabelo avermelhado.



Figura 5-1- Personagem procurada - Juca

Para a recuperação das imagens que contém o objeto Juca, foram criadas 3 consultas: uma para sua característica mais marcante, a camisa, outra para o contorno de seu rosto e uma outra com as características de seu cabelo.

Utilizando uma retroalimentação das 3 consultas citadas acima, o sistema obteve um índice de precisão de 99,29% e o índice de revocação ficou em 99,64%. A tabela abaixo mostra os resultados

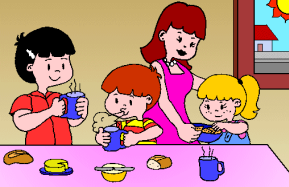



Tabela 5-2 - Resultados da busca do personagem Juca

Resultados das buscas	Imagens Relevantes	Imagens Não Relevantes	Total
Resultados Positivos	840	3	843
Resultados Negativos	3	1320	1323
Total	843	1323	4332

Com estes dados temos a sensibilidade da busca com 99,64%, a especificidade com 99,77%, o valor preditivo positivo com 99,64% o valor preditivo negativo com 99,7%.

Embora o objeto tenha uma característica muito marcante, a camisa que isolada obteve um índice de precisão de 98,29%, a retroalimentação serviu para incluir nos resultados relevantes imagens em que Juca estava vestindo outra camisa ou ela estava ocluída. Os resultados negativos

ocorreram em imagens que os contornos do rosto se assemelham com a do objeto em questão ou objetos semelhantes à camisa estão presentes.

Relevantes	Não Relevantes
 <p data-bbox="326 657 789 732">Figura 5-2 - Imagem Recuperada mesmo com o objeto parcialmente ocluído</p>	 <p data-bbox="924 602 1398 678">Figura 5-3 Imagem falsa recuperada devido a semelhança dos traços do cabelo</p>
 <p data-bbox="370 993 769 1068">Figura 5-4 - Imagem recuperada pelo cabelo do personagem</p>	 <p data-bbox="938 1052 1390 1127">Figura 5-5 Imagem falsa recuperada pelas listras vermelhas e amarelas</p>

A estrutura final da consulta pode ser visualizada pelo grafo da Figura 5-6.

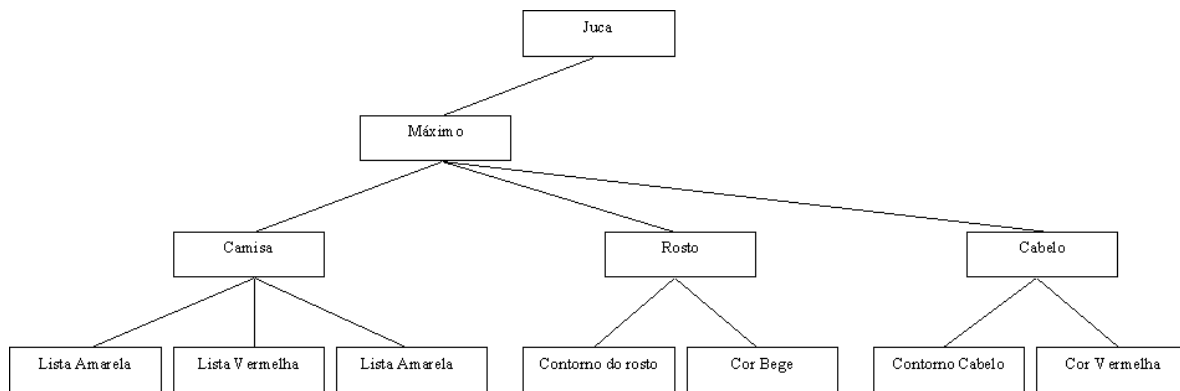


Figura 5-6 - Grafo da retroalimentação do objeto Juca

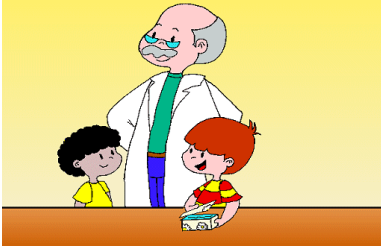
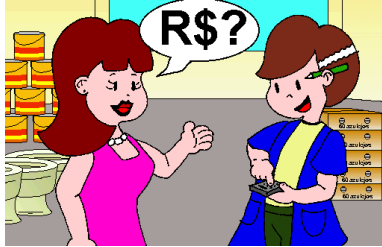

Resultados intermediários

- Sentença – *Camisa*

Esta sentença foi montada utilizando as informações das cores predominantes das listras da camisa do personagem Juca, sendo que essas listras estão “em cima” e “muito perto” uma da outra, semelhante ao ilustrado na Figura 2-7.

A camiseta deste personagem é uma característica muito marcante. Por este motivo, o índice de precisão ficou em 98,29%, 0,98% de falsos positivos e os falsos negativos ficaram em 0,73%. Já o índice de revocação ficou em 99,26%.

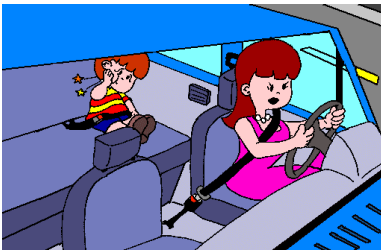
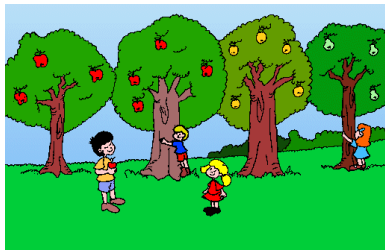
Os falsos negativos ocorreram quando a personagem está parcialmente ocluída.

Acerto	Falso Positivo	Falso Negativo
 <p data-bbox="224 1241 602 1318">Figura 5-7 – Resultado correto com 100% de semelhança</p>	 <p data-bbox="618 1241 997 1318">Figura 5-8 – Resultado falso positivo com índice 53%</p>	 <p data-bbox="1016 1255 1401 1333">Figura 5-9 – Falso negativo com 48% de semelhança</p>

- Sentença – *Camisa e Rosto*

A junção entre as sentenças camisa e do contorno da área, de cor bege, do rosto do personagem Juca, com a condição de que uma esteja “em cima” e “muito perto” da outra, formaram essa sentença.

O índice precisão dessa sentença ficou em 97,26%, os falsos positivos 2,74% e não houve resultados falsos negativos, com isso o índice de revocação ficou em 100%.

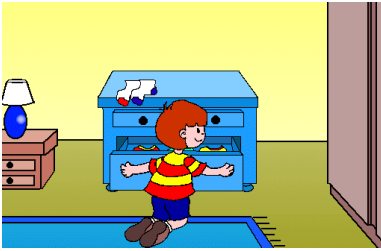


Acerto	Falso Positivo	Falso Negativo
 <p data-bbox="224 604 602 676">Figura 5-10 - Resultado positivo com índice de 90%</p>	 <p data-bbox="618 604 1000 676">Figura 5-11 – Resultado falso positivo com índice de 60%</p>	

- Sentença – *Camisa e cabelo*

Esta sentença foi montada utilizando as informações das listras da camiseta (as cores e as suas ordens) e o contorno da área, de cor vermelha, da cabeça, sendo esse “em cima” das listras.

A precisão ficou em 83,04%, os falsos positivos em 12,87% e os falsos negativos em 5,26%. A revocação ficou em 95,30%.

A maioria dos falsos negativos ocorreram quando na imagem existem objetos da cor das listras da camiseta, uma vez que essa sentença só procura pela cor e ordem, ignorando a forma ou se a personagem está de costa.

Acerto	Falso Positivo	Falso Negativo
		
<p>Figura 5-12 – Um dos resultados positivos com 84% de semelhança</p>	<p>Figura 5-13 – Resultado falso positivo com índice de 52%.</p>	<p>Figura 5-14 – Resultado falso negativo com índice de 49%</p>

Objeto Laura

O objeto Laura é uma personagem feminina caracterizada, principalmente, pelo seu macacão azul, a blusa rosa por baixo do macacão, e o cabelo loiro e longo preso.

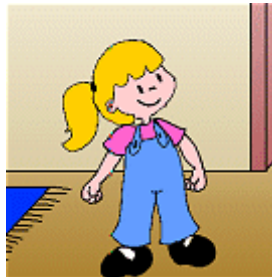


Figura 5-15 Personagem procurada - Laura

As 3 consultas principais para a personagem Laura foram; Cabeça (Rosto e Cabelo), Macacão e Camisa. O refinamento com essas 3 consultas resultou em um índice de precisão de 85,71% e revocação de 87,80%. A tabela abaixo mostra os resultados da busca.

Tabela 5-3 - Resultados da busca do personagem Laura

Resultados das buscas	Imagens Relevantes	Imagens Não Relevantes	Total
Resultados Positivos	540	15	555
Resultados Negativos	60	1551	1611
Total	600	1566	4332

A sensibilidade da busca ficou em 90,00%, a especificidade em 99,04%, o valor preditivo positivo em 97,30% o valor preditivo negativo em 96,28%.

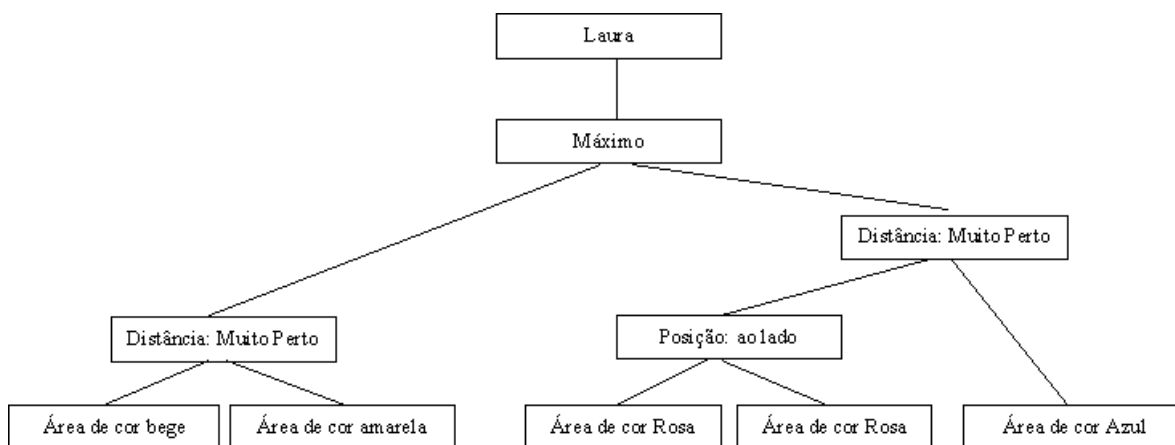
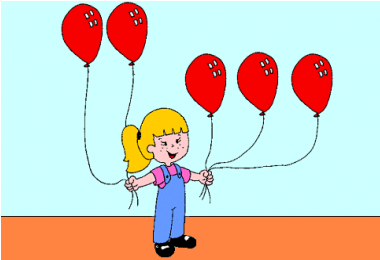
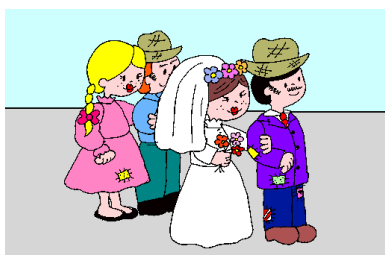



Figura 5-16 - Grafo do objeto de busca para Laura

- Sentença – *Rosto e Cabelo (Cabeça)*

Esta sentença foi construída utilizando as informações da cor do rosto e da cor do cabelo, estando um “em cima” do outro a uma distância relativa “muito perto” e a área com a cor do rosto “menor” que a área do cabelo.

O índice de precisão ficou em 74,87%, o de resultados falsos positivos em 7,18% e os falsos negativos em 17,95%. O índice de revocação ficou em 80,66%.

Acerto	Falso Positivo	Falso Negativo
		
<p>Figura 5-17 – Resposta correta com 100% de semelhança.</p>	<p>Figura 5-18 - Resultado falso positivo com índice de 70%.</p>	<p>Figura 5-19 – Resultado falso negativo com índice de 26%.</p>

- Sentença - *Macacão, rosto e cabelo*

Para formar esta sentença foram utilizadas as informações da cor do macacão e as informações da cor do rosto e cabelo, estando um “em cima” do outro a uma distância relativa “muito perto”.

A precisão ficou em 75,00%, os falsos positivos responderam por 7,22% e os falsos negativos em 17,78%. O índice de revocação ficou em 80,84%.

Os falsos positivos continuaram com o mesmo índice da sentença anterior e uma melhora insignificante nos falsos negativos.

- Sentença – Macacão e Camisa

Esta sentença utilizou a mesma notação para o macacão e foi acrescida a informação da cor da camiseta do objeto à sentença.

O índice de precisão da sentença ficou em 79,35% de acerto, 2,72% de falsos positivos e 17,93% de falsos negativos. A revocação ficou em 81,56%.

Objeto Vovô

Este é um personagem masculino, o avô de Juca e Laura, caracterizado pela calça azul e um jaleco branco e longo por cima (primeira sentença). A personagem tem bigode e cabelos acinzentados e é calvo (segunda sentença).



Figura 5-20 - Personagem Vovô

Fazendo o refinamento nas duas principais consultas utilizadas para definir o personagem, chegamos a um resultado de 84,38% de precisão e uma revocação de 90,00%.

A sensibilidade da busca ficou em 96,43%, a especificidade em 99,71%, o valor preditivo positivo em 93,10% o valor preditivo negativo em 99,86%.

A tabela abaixo mostra os resultados da busca.

Tabela 5-4 - Resultados da busca do personagem Vovô

Resultados das buscas	Imagens Relevantes	Imagens Não Relevantes	Total
Resultados Positivos	81	6	87
Resultados Negativos	3	2076	2079
Total	84	2082	4332

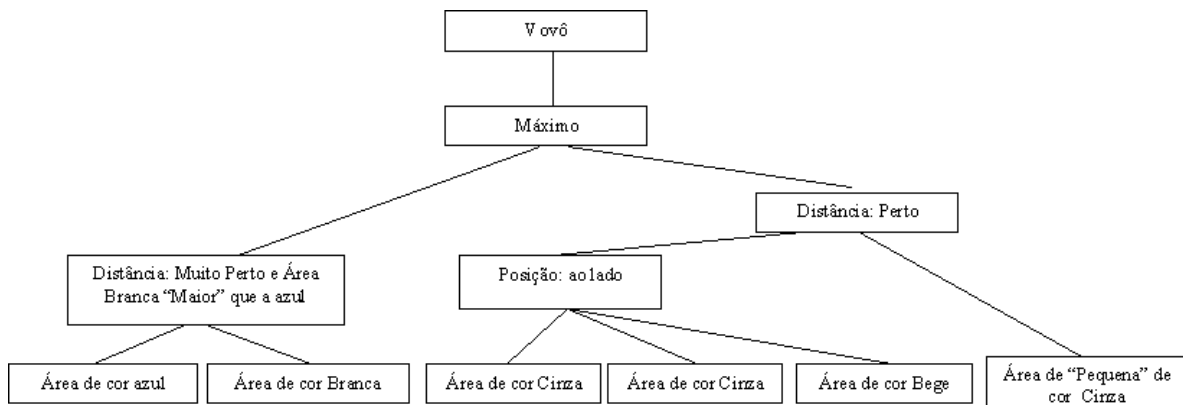


Figura 5-21 - Grafo da busca do personagem Vovô

Resultados intermediários

- Sentença – *Calça e Camisa*

A primeira sentença do objeto utilizou as informações das cores da calça e da camisa a uma distância “muito perto” e que a cor da calça estivesse “em cima” da cor da camisa e que a área da cor da camisa fosse “pouco maior” que a área da cor da calça.

Essa sentença teve um índice de acerto de 82,14%, falsos positivos em 13,02% e falsos negativos em 4,84%. A revocação ficou em 85,19%.

- Sentença – *Cabeça*

Na montagem desta sentença foram escolhidas a cor do cabelo (duas partes) e a cor da parte calva da cabeça com a relação de que a parte calva ficasse entre as duas cores do cabelo. As áreas do cabelo devem ser um “pouco menor” que a área calva.

Esta sentença resultou em índices de acertos inferiores ao anterior, ficando com 63,33% de acertos, 35,03% de falsos negativos e 1,64% de falsos positivos. A revocação ficou em 70,37%.

Objeto Árvore

Para recuperar imagens em que constem uma árvore de copa fechada foram montadas duas consultas principais:

- Tronco – caracterizado por uma área retangular de cor marrom.
- Copa – caracterizada por uma área oval de cor verde

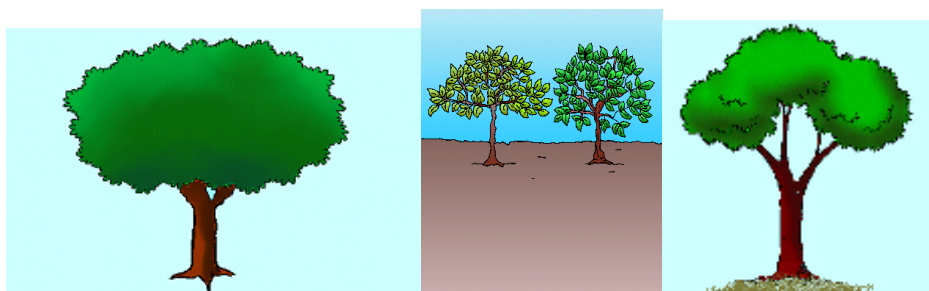


Figura 5-22 - Algumas figuras de árvores presente na base de dados

No refinamento deste objeto, foi obtido um índice de precisão 76,10% e a revocação foi de 98,73%.

A sensibilidade da busca ficou em 99,36%, a especificidade em 91,34%, o valor preditivo positivo em 75,98% o valor preditivo negativo em 99,81%.

A tabela abaixo mostra os resultados da busca.

Tabela 5-5 - Resultados da busca do personagem Árvore

Resultados das buscas	Imagens Relevantes	Imagens Não Relevantes	Total
Resultados Positivos	465	147	612
Resultados Negativos	3	1551	1554
Total	468	1698	4332

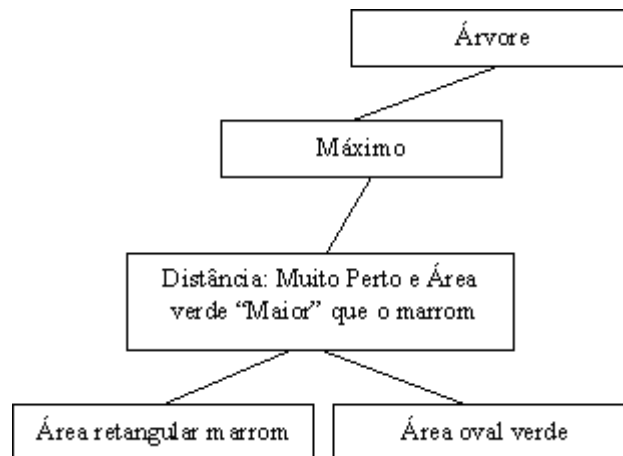


Figura 5-23 - Grafo do objeto de busca árvore

Resultados intermediários

- Sentença – *Tronco e Copa verde*

Na montagem desta sentença de busca foram escolhidas uma área de cor marrom e uma área de cor verde. Esta sentença teve um índice de acerto 73,79%, os falsos positivos ficaram 9,68% e os falsos negativos em 16,53%.

- Sentença – *Tronco e Copa com informações espaciais*

Na montagem desta sentença de busca foram escolhidas uma área de cor marrom e uma área de cor verde, sendo que uma deve estar “muito perto” da outra. Esta sentença teve um índice de acerto 61,85%, os falsos positivos ficaram 7,23% e os falsos negativos em 30,92%.

5.1 Comparações

Em processamento digital de imagem, embora a extração da forma dos objetos seja uma tarefa difícil, ela é uma poderosa ferramenta para sua identificação (Chang e Chan, 2000). Nós, humanos, por exemplo, podemos reconhecer características dos objetos somente pela sua forma. A teoria de linguagens formais nebulosas pode ser utilizada para simular o funcionamento do sistema visual humano (Rocha 1997,1999 e Serapião 1996,1997). O processamento no nível da

retina pode ser simulado por vários tipos de filtros e algoritmos para extração de formas, enquanto que filtros de segmentação podem simular circuitos pré-corticais.

Recentemente, a forma dos objetos tem atraído muito a atenção nas pesquisas sobre sistemas de buscas de imagens. Duas representações de formas são usualmente empregadas nos sistemas de buscas: os descritores de forma baseados em regiões e os descritores de forma baseados em contorno (Bober, 2001; Sikora, 2001; Kim e Kim, 2000; Chans et al, 1998; Bimbo et al, 1996; Takahashi et al, 2000; Zhou e Huang, 2001; Zhou, Rui e Huang, 1999). Dois desses sistemas (Bober, 2001 e Zhou e Huang, 2001) foram usados para discutir comparativamente os resultados do presente projeto.

Foram escolhidos porque na época eram os únicos que permitiam a inserção de figuras em suas base de dados.

Bober(2001) apresenta o descritor baseado em contorno. Esse descritor é denominado descritor CSS (curvature scale-space), pois utiliza o princípio de evolução da forma. A partir de um contorno fechado, é calculada a curvatura da forma e são selecionados os pontos de cruzamento. Com esses pontos, gera-se uma nova imagem que caracteriza o comportamento da forma em uma próxima resolução. Um índice CSS é usado para emparelhar e indicar o peso dos picos superiores, chamados de imagens CSS.

Inicialmente, é extraído o contorno da figura a ser processada e, em seguida, calculada a curvatura e selecionados todos os pontos de cruzamento (Figura 5-25). Na Figura 5-26, é mostrado o contorno da figura sendo processada, após 24 iterações. Nota-se a grande diminuição dos pontos de cruzamento. A Figura 5-30 mostra o contorno ao final do processamento.

Esse processo de cálculo de curvatura é semelhante ao processo de identificação por pontos significativos, usado no presente projeto (Serapião, 1996). Por isso o método de Bober, (2001) poder ser facilmente utilizado como uma ferramenta auxiliar no presente projeto.

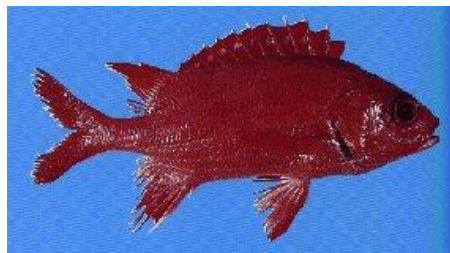


Figura 5-24 - Imagem a processar

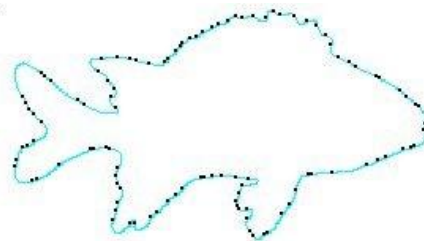


Figura 5-25 – Contorno fechado e a curvatura da forma com os pontos de cruzamento

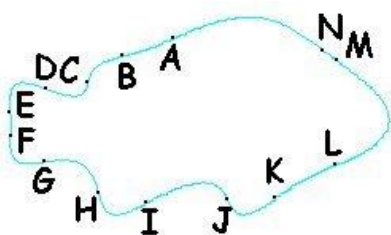


Figura 5-26 – Evolução da forma: contorno após 24 iterações



Figura 5-27 – Comportamento da curvatura após 24 iterações.

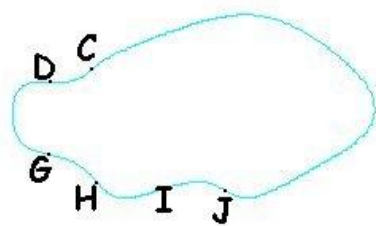


Figura 5-28 - Evolução da forma: contorno após 35 iterações

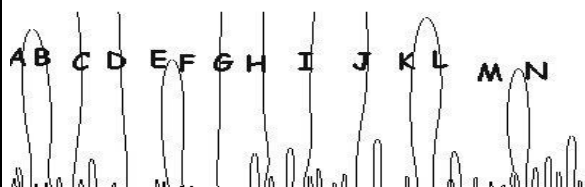
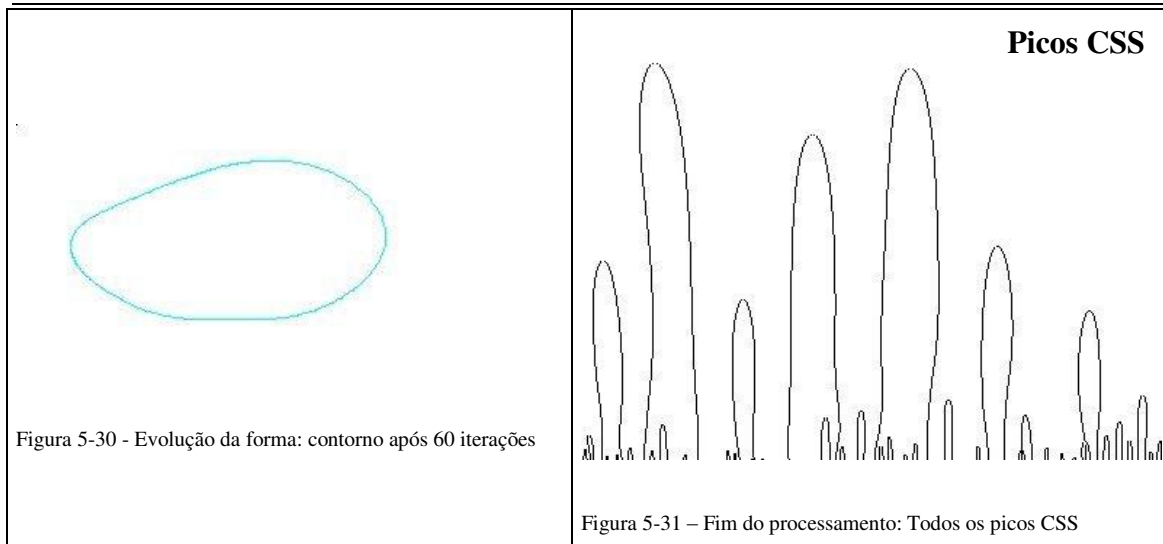
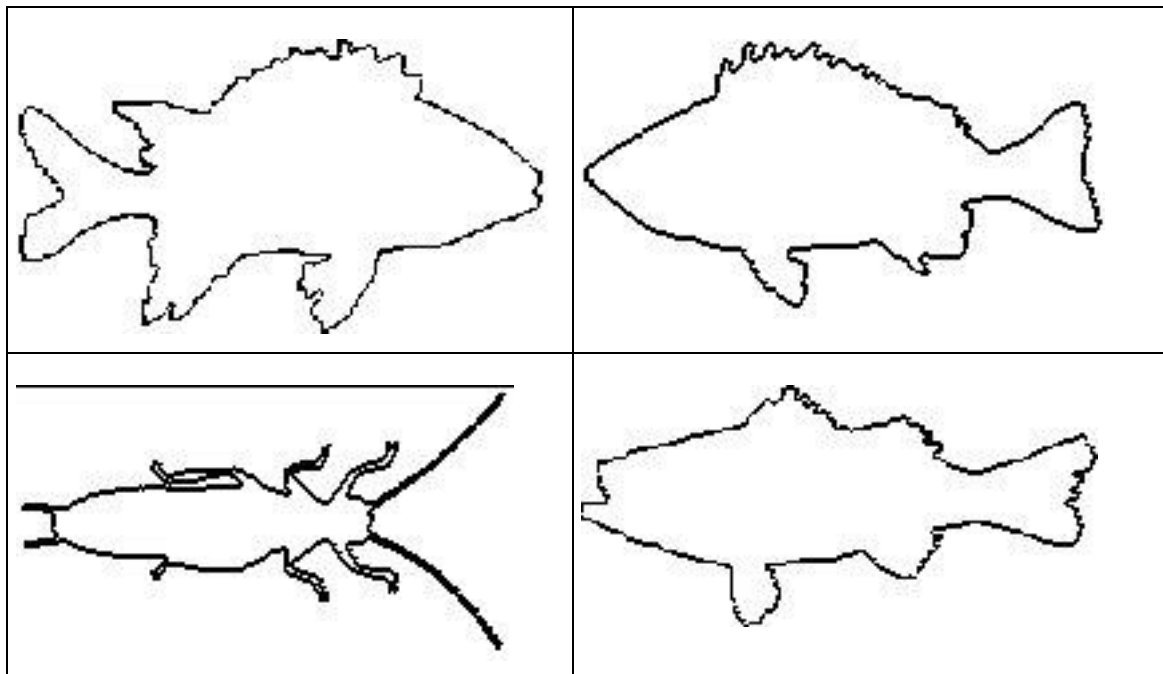


Figura 5-29 – Comportamento da curvatura após 35 iterações.



A seguir são mostrados os contornos semelhantes ao da Figura 5-24



Em Zhou e Huang(2001) um descritor foi implementado baseado no contorno (é utilizado o “water-filling” que encontra o contorno dos objetos por um método similar a uma “inundação por água”) e fundamentado em características estruturais como “MaxFillingTime” (“Filling time” é o tempo para preencher as bordas conexas), “MaxForkCount” (“Fork Count” é o número total de ramificações onde ocorreram bifurcações).

		1	2	3	4		
					5		
		9	8	7	6		
		10			7		
		11			8		
		12	11	10	9	10	

Figura 5-32 – Imagem a rotulada após a execução do algoritmo water-filling.

O algoritmo water-filling faz uma varredura nos contornos da imagem e “preenche com água” o primeiro pixel do contorno que tenha pelo menos dois vizinhos, sendo que esse recebe o rótulo “1”. O fluxo de água flui ao longo do contorno e cada pixel vizinho do primeiro recebe um número, em ordem crescente. Quando existir mais de um caminho, o fluxo de água se dividirá (fork). Para implementar o algoritmo foi suposto que (1) existe água em quantidade ilimitada desde o ponto inicial; (2) O fluxo de água possui uma velocidade constante em todas as direções; (3) O fluxo de água só pára quando encontra o ponto final do contorno ou todas as direções já tenham sido preenchidas. Como os canais do fluxo de água, várias outras informações são extraídas (Tempo de preenchimento, Quantidade de bifurcações, Quantidade de laços, Quantidade de água (o número do último pixel rotulado), etc.) e é construído um vetor de características primitivas. A partir das características primitivas são construídas as características estruturais.

A complexidade deste algoritmo é linear, proporcional ao número de pontos do contorno na imagem.

A Figura 5-32 ilustra o preenchimento do contorno de uma imagem da letra “a”. Pode-se notar nessa imagem as características primitivas, como por exemplo, o tempo de preenchimento (12), a quantidade de bifurcações (3), a quantidade de laços (1) e a quantidade de água utilizada (18).

5.1.1 Comparação de eficiência e velocidade

Um sistema de busca eficiente tem que obedecer as seguintes características (Jain e Vailaya, 1998 e Kim e Kim, 2000):

Precisão: Deve ser preciso, as imagens encontradas têm que se assemelhar ao objeto em questão.

Velocidade: Como os bancos de dados de imagens tipicamente têm milhares de imagens, o processamento deve ser em tempo-real, isto é, os resultados da busca devem ser rápidos.

A Figura 5-33, Figura 5-34 e Figura 5-35 mostram os resultados obtidos das buscas de alguns objetos (V=50).



Figura 5-33 – Alguns resultados do objeto Juca



Figura 5-34 - Alguns resultados do objeto Árvore



Figura 5-35 – Alguns resultados do objeto Laura

5.1.1.1 Precisão:

A precisão de um sistema de busca é definida por (Fuh, Cho e Essig, 2000; Hsieh e Fan, 2001; Lecce e Guerriero, 1999):

(1) A precisão da busca de um objeto Q é definida como

$$\eta_q = \begin{cases} (\mu/V) \times 100\% & \text{se } V \leq W \\ (\mu/W) \times 100\% & \text{se } V > W \end{cases}$$

Equação da Precisão da busca

onde W é o número de imagens similares de Q encontrados na base de dados por um método subjetivo, V é o número de imagens similares de Q encontradas na base de dados pelo sistema e

μ é o número de imagens similares encontradas por ambos. Foi utilizada essa definição de precisão porque ela possibilita analisar os resultados por faixa de precisão, uma vez que V são as imagens ordenadas, com os maiores valores de pertinência acima do limiar escolhido pelo usuário. Para esta comparação V será de 10, 20, 30, 40 ou 50.

(2) A precisão global é definida pela média das precisões para G imagens, isto é:

$$\eta = \frac{1}{G} \sum_{q=1}^G \eta_q$$

Equação da Precisão média da busca

Foram realizadas procura de 9 (nove) objetos em uma base de dados com 960 imagens para avaliar a precisão do sistema, baseando-se nas equações acima. A média W dos 9 objetos ficou em torno de 195.33.

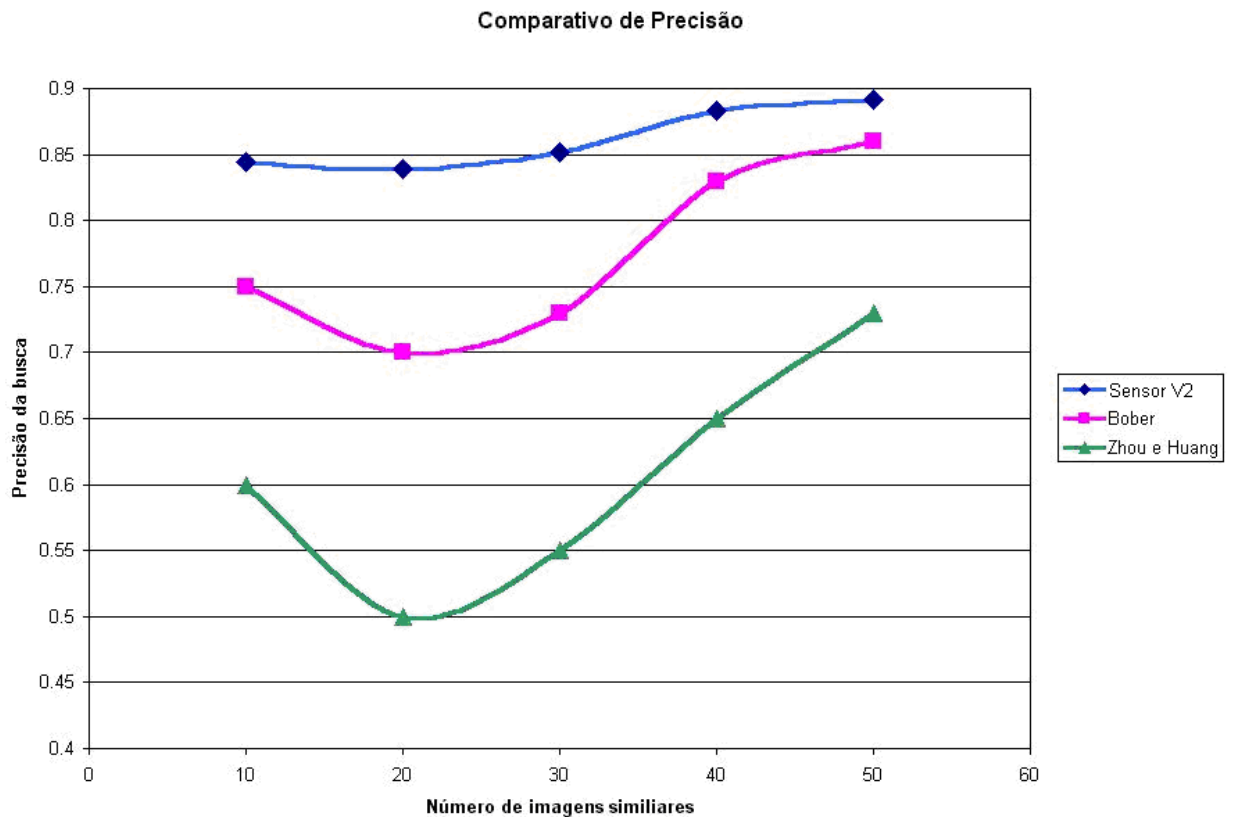


Figura 5-36 - Precisão média das buscas dos 3 sistemas

A Figura 5-36 mostra um gráfico ilustrando a precisão dos 3 sistemas como uma função do número da demanda de imagens similares retornada pelo sistema.

5.1.1.2 *Tempo de execução*

Geralmente, a velocidade da busca é medida pela média do tempo de busca (Wang, Li e Wiederhold, 2001). Esse tempo médio depende principalmente do tamanho da base de dados de imagens, das condições do sistema de software e hardware, do tamanho das imagens e do número de processos de comparação entre os objetos a serem procurados e a imagem sendo processada. Os testes foram realizados em um PC utilizando a plataforma Intel Pentium III 500Mhz com memória principal de 128Mb DRAM sobre o sistema operacional Microsoft Windows XP. A plataforma de desenvolvimento foi o Microsoft Visual Basic (interface com usuário) e o Visual C++ (implementação dos filtros). Nossa base de dados de imagem possui aproximadamente 22.000 imagens e para os testes foram escolhidas aleatoriamente 960 imagens e 9 objetos a serem procurados. A figura abaixo mostra o gráfico com a comparação da velocidade dos sistemas.

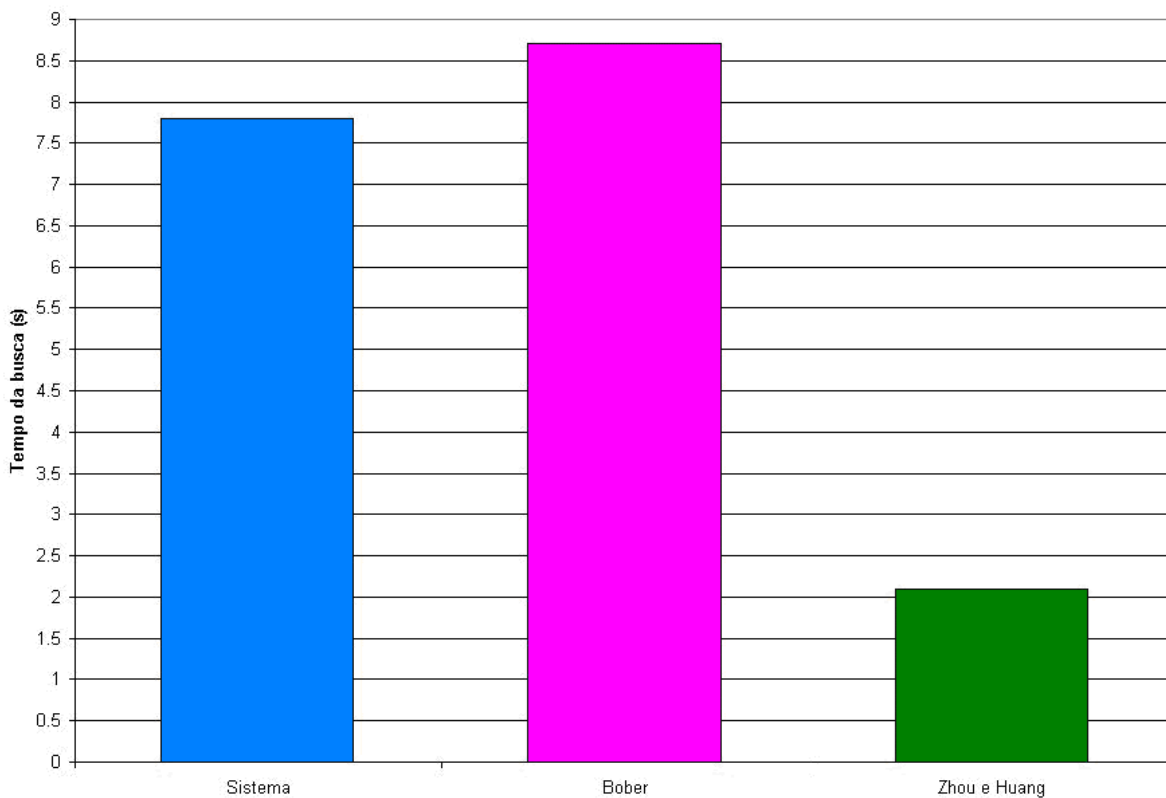


Figura 5-37 - Comparação de tempo entre os 3 sistemas

De acordo com o gráfico podemos concluir que o sistema Sensor V2 possui um desempenho superior de aproximadamente 34% em relação ao sistema de Zhou e Huang(2001). Em contrapartida, o sistema de Zhou e Huang(2001) foi 3,7 vezes mais rápido que o nosso sistema. Comparando o nosso sistema com o de Bober, 2001, a desempenho do nosso ficou 10,5% superior e a velocidade em 11,5% superior. Esses valores ficaram próximos devido à semelhança entre os dois algoritmos, conforme já explanado. O desempenho superior do nosso sistema deve-se ao fato de levar em consideração, além do contorno, as cores dos objetos a serem procurados na base de dados do sistema ENSCER.

6 Conclusões

O sistema se mostrou eficiente tanto no reconhecimento de objetos característicos quanto genéricos, como podemos observar nos índices precisão de 99,29% e de revocação de 99,64% para o personagem característico Juca e para um objeto mais genérico como, por exemplo, a árvore, que apresentou precisão de 75,73% e revocação de 98,73%. Conforme já relatado, o valor da revocação muito superior ao da precisão significa que o sistema retornou vários resultados falsos positivos.

Em relação ao desempenho, o sistema demonstrou um desempenho superior em relação a alguns dos disponíveis na literatura (Bober, 2001) e um pouco inferior a outros (Zhou e Huang, 2001). O desempenho talvez pudesse ser melhorado caso utilizássemos um gerenciador de banco de dados específico para imagens, como por exemplo, o modelo de banco de dados, voltado para o desenvolvimento de aplicações espaciais e geográficas, proposto em (Ozsu et al, 2000).

Com a TCP, Zadeh(2001) propõe que é possível realizar uma grande variedade de tarefas físicas e mentais sem a necessidade de medições ou cálculos precisos. Os valores dos atributos são granulados, e cada grânulo reúne um grupo de valores (pontos, objetos) por similaridade, proximidade ou função, onde os grânulos de uma distância poderiam ser definidos como muito perto, perto, médio, longe, muito longe, dentre outros. Quando esses grânulos têm limites imprecisos são chamados F-Grânulos. A percepção humana é organizada a partir de F-Grânulos e descrita, na maioria das vezes, em linguagem natural.

Neste trabalho, a análise de uma imagem utilizou a questão do tipo:

“Localização de X em relação a Y”

e como solução à questão obtivemos dois tipos de proposições:

a-) sobre as relações de distância entre os objetos que compõe a cena {Muito Perto, Perto, Média, Longe, Muito Longe};

b-) sobre as direções entre esses mesmos objetos {Norte, Sul, Leste, Oeste, Noroeste...}.

Portanto, o fundamento da teoria computacional das percepções proposta por Zadeh foi evidenciada neste trabalho, já que utilizando esse conceito conseguimos obter imagens semelhantes, a partir de uma sentença montada pelo usuário.

Os resultados obtidos apóiam o modelo que vem sendo desenvolvido no laboratório com intuito de melhor entender o sistema visual humano.

Através deste trabalho foi possível também ampliar o sistema SENSOR (Serapião, 1996; Rocha 1997; Rodella, 2002) que até então possuía os seguintes níveis de capacidade:

- Nível I: Extração de informações básicas sobre cores, luminância e contraste em uma imagem.
- Nível II: Reconhecimento dos atributos primários da imagem, tais como linhas, ângulos, descontinuidades de linhas, etc.
- Nível III: Reconhecimento de objetos a partir do conjunto de atributos reconhecidos.
- Nível IV: Análise da relação de direção e distância entre os objetos bem conhecidos presentes numa cena prototípica.
- Nível V: Geração de uma cena prototípica média a partir de um conjunto de cenas escolhidas.

Com a realização deste trabalho, foram aumentados os níveis

- Nível VI: Montagem de objetos a serem procurados.
- Nível VII: Recuperação de imagens que contém os objetos semelhantes aos montados.

6.1 Sugestões de trabalhos futuros

Em um próximo trabalho, poderão ser desenvolvidas as seguintes modificações:

- desenvolver uma interface mais amigável com o usuário, utilizando de técnicas de usabilidade;

-
- portar o sistema para o ambiente Internet:
 - utilizando o navegador como interface com o usuário;
 - pesquisando e indexando as imagens automaticamente;
 - melhorar o algoritmo de busca para reconhecer faces humanas e/ou expressões faciais;
 - permitir a busca de objetos não só em imagens estáticas como também em arquivos de vídeo.

7 Bibliografia

Alegre, L., Rocha, A.F., Morooka, C.K. Intelligent approach rod pumping problems. SPE 26253, pp 249-255, 1993a.

Alegre, L., Morooka, C.K., Rocha, A.F. Intelligent approach rod pumping problems. SPE 26516, pp 97-107, 1993b.

Aslandogan, Y. A. e Yu, C. T. “Techniques and systems for image and video retrieval”, IEEE transactions on Knowledge and Data Engineering. 11(1), 56-63, 1999.

Bhatnagar, S.C., Mandybur, G.T., Buckingham, H.W., Andy, O.J. “Language Representation in the Human Brain: Evidence from Cortical Mapping”. In: Language and Brain. 2000; 74: 238-259.

Bimbo, A. Del, Pala, P., Santini, D., 1996. Image retrieval by elastic matching of shapes and image patterns, in: Proceedings of the IEEE Multimedia, Hiroshima, Japan, pp 215-218.

Bober, M. MPEG-7 visual shape descriptors, IEEE Trans. Circuits Systems Video Technol., 11, pp. 716-719. ,2001

Bordogna, G. Pictorial indexing for an integrated pictorial and textual information retrieval environment. Information Services & Use, v. 16, p. 165-173, 1990.

Cabeza, R., Nyberg, L. “Imaging cognition II: an empirical review of 275 PET and fMRI studies”. In: Journal Cognitive Neuroscience.12: 1-47, 2000.

Carson, C., Thomas, M., e et al. “BlobWorld: a system for region-based image indexing and retrieval”. Proceedings of the third Internacional Conference on Visual Information Systems, 1999.

Cattell, R.G.G., Barry, D. e et al. “The Object Data Standard: ODMG 3.0”. Morgan Kaufmann, 2000.

Chang, C.C., Chan, Y.K.. A fast filter for image retrieval based on color-spatial features. In: SEMA2000. Baden-Baden, German, pp. 47-51, 2000.

Chakravarthy, A. S. Toward semantic retrieval of pictures and video. Proc. Riao'94, Intelligent Multimedia Information Retrieval Systems and Management, New York, p. 676-686, 1994.

Chans, Y., Lei, Z.B., Lopresti, D., Kung, S.Y. A feature-based approach for image retrieval by sketch, Proc. SPIE 3430 pp. 72-82, 1998.

Chomsky, N. "Aspects of the Theory of Syntax". MIT Press. Cambridge, 1965.

Chugani, H.T. "Metabolic Imaging: A window on brain development and plasticity". In: The Neuroscientist. 5: 29-40, 1999.

Rodella, E. C. "Metodologia para Análise de Cenas e Geração de Cenas Prototípicas Utilizando a Teoria Computacional das Percepções." Tese de Mestrado, Universidade Estadual de Campinas – Faculdade de Engenharia Elétrica – Departamento de Computação e Automação. 2002.

Flickner, M., Sawhney, H. e et al. "Query by image and video content: The QBIC system". IEEE Computer, 28(9), 23-31, 1995.

Fuh, C. S., CHO, S. W., Essig, K. Hierarchical color image region segmentation for content-based image retrieval system, IEEE Trans. Image Process. 9, pp 156-162, 2000.

Guilherme, I.R. "Sistema Jargão – Um sistema para Análise de Base de Dados em Linguagem Natural". Tese de Doutorado, Universidade Estadual de Campinas – Faculdade de Engenharia Elétrica – Departamento de Computação e Automação. 1996.

Gong, Y., "Intelligent Image Databases. Towards Advanced Image." Kluwer Academic Publishers, 1998

Gonzalez, S. M. “Modelo para o Gerenciamento de Banco de Dados de Imagens” Tese de Doutorado, Universidade Estadual de Campinas – Faculdade de Engenharia Elétrica – Departamento de Computação e Automação. 2003.

Henrich, A. e Robbert, G. “MARS: A retrieval service for multimedia authoring environments”. ADBIS-DASFAA Symposium. 88-98, 2000.

Hirata, K., Hara, Y., Li, W.S. e Candan, K.S. “Facilitating multimedia database exploration through visual interfaces and perceptual query reformulations”. Proceedings of the 23^a VLDB conference, 538-547, Athens, Greece, 1997.

Hopcroft, J.E., and Ullman, J.D. Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Reading, MA, 1979.

Hsieh, I. S., Fan, K. C. Multiple classifiers for color flag and trademark image retrieval, IEEE Trans. Image Process. 10, pp 938-950, 2001.

Jain, A. K., Vailaya, A. Shape-based retrieval: a case study with trademark image database, Pattern Recognition 31, pp 1369-1390, 1998.

Kanwisher, N., Wojciulik, E. “Visual attention: Insights from brain imaging”. In: Nature Reviews: Neuroscience.1: 91-100, 2000.

Kim, H. K., Kim, J. D. Region-based shape descriptor invariant to rotation, scale and translation, Signal Processing: Image Communication 16, pp 87-93, 2000.

Klir, G.J., Yuan, B. Fuzzy Sets and Fuzzy Logic Theory and Applications. Prentice Hall PTR, New Jersey, 1995.

Lecce, V. D., Guerriero, A. Na evaluation of the effectiveness of image features for image retrieval, J. Visual Commun. Image Representation 10, pp. 351-362, 1999.

Lee, D., Barber, R., Petkovic D. e et al. “Complex queries for a query-by-content image database.” Technical Report, IBM Research Division, 1994.

Leung, C. H. C.. Architecture on an image database system. *Information Services & Use*, v. 10, p. 391-397, 1990.

Li, W.S., Candan, K.S., Hirata, K. e Hara, Y. "A hybrid approach to multimedia database systems through intergration of semantics and media-based search." *WWCA'97, Proceedings*, 1274, 182-197, Springer-Verlag, 1997.

Lucchini, F. L. P. "Controle do computador usando movimentos do corpo, identificados por um adesivo, capturados por uma câmera de vídeo webcam" Tese de Mestrado, Universidade Estadual de Campinas – Faculdade de Engenharia Elétrica – Departamento de Computação e Automação. 2001.

Ma, W.Y e Manjunath, B.S. "NETRA: a toolbox for navigating large image databases." *Multimedia Systems*, 7(3):184-198, 1999.

Maguire, E.A., Burgess, N., Donnett, J.G., Frackowwiak, R.S.J., Fith, C.D., O'Keefe, J. "Knowing where and getting there: A human navigation network". In: *Science*. 280: 921-924, 1998.

Mandal, M. K., Aboulnasr, T., Panchanathan, S. Image Indexing Using Moments and Wavelets. *IEEE Transactions on Consumer Electronics*, v 42, n. 3, p. 557-565, Agosto 1996.

Mellet, E., Txourio-Mazoyer, N., Bricogne, S., Mazoyer, B., Kosslyn, S.M., Denis, M. "Functional anatomy of thigh-resolution visual mental imagery". In: *Journal of Cognitive Neuroscience*. 12: 98-109, 2000.

Menezes, P.B. *Linguagens Formais e Autômatos*. Sagra Luzzatto, Porto Alegre, 4a. Edição, 2001

Mizumoto, M. et al. "N-Fold Grammar". In: *Information Sciences*. 5: 22-32, 1973.

Mukherjea, S., Hirata, K. e Hara, Y. "AMORE: a World Wide Web Image Retrieval Engine." *World Wide Web*, Baltzer, 2(3):115-132, 1999.

Negoita, C.V., Ralescu, D.A. *Applications of Fuzzy Sets to Systems Analysis*. John Wiley & Sons, New York, 1975.

Nepal, S. e Ramakrishna, M.V. "Query processing issues in image (multimedia) databases. ICDE, 22-29, 1999.

Nepal, S., Ramakrishna, M.V. e Thom, J.A., "A fuzzy object query language (FOQL) for image databases" Proceedings of the 6th International Conference on Database Systems do advanced applications, 1999.

O'Craven, K.M., Kanwisher, N. "Mental imagery of faces and places activates corresponding stimulus-specific brain regions". In: *Journal of Cognitive Neuroscience*. 12: 1013-1023, 2000.

Ogle, V. E., Stonebraker, M. "Chabot: retrieval from a relational database of images". *IEEE Computer*, 28(9):40-48, Setembro 1995.

Ozsu, M. T. e et al. "DISIMA: A distributed Image Database Management System," Technical Report STR181014, university Alberta, 2000.

Pressman, R.S. *Engenharia de Software*. Makron Books, São Paulo. 453-457, 1995.

Raij, T. "Patterns of brain during visual imagery of letters". In: *Journal of Cognitive Neuroscience*. 11: 282-299, 1999.

Ramakrishna, M.V. e Nepal, S. "Similarity query processing in image databases. Technical Report 2000/61, School of Computer Science and Software Engineering, Monash University, 2000.

Riesenhuber, R., Poggio, T. "Hierarchical models of object recognition in cortex". In: *Nature Neuroscience*. 2: 1019-1025, 1999.

Rocha, A. F. "O cérebro na escola". EINA, Jundiaí, 2000

Rocha, A.F. *O cérebro: Um breve relato de sua função*. EINA, Jundiaí, 1999.

Rocha, A. F., A. B. Serapião; A. Rondó, E. C. Rodella; F. Lucchini; M. P. Rebello. Correlating Intelligence and Brain Activity. Proc. 6th Ann. Meeting Cognitive Neurosciences Society, Washington, D.C, USA, 1999.

Rocha, A.F., Rebello, M.P., Miura, K. "Toward a Theory of Molecular Computing". In: Information Sciences; 106: 123-157, 1998.

Rocha, A. F.. The brain as a symbol processing machine. Progress in Neurobiology. v. 53, no.2, pp. 121-198, 1997.

Rocha, A.F. "The brain as a symbol processing machine". In: Progress in Neurobiology. 53:121-198, 1997.

Rocha, A.F. "Neural Nets: A theory for brains and machines". In: Lectures Notes in Artificial Intelligence, Springer-Verlag, Berlin, vol. 638, 1992.

Rocha, A.F., Guilherme, I.R., Theoto, M., Miyadahira, A.M.K., Koizumi, M.S. "A Neural Net for Extracting Knowledge from Natural Language Data Bases". In: IEEE Transactions on Neural Networks, vol. 3, No. 5, 1992.

Rocha, A.F., Françoso, E., Balduino, M. A. "Neural Languages". In: Fuzzy Sets and Systems. 3/1, 11-35, 1980.

Rui, Y., Huang, T., e Mehrotra, S. "Content-based image retrieval with relevance feedback im MARS". Proceedings of IEEE International Conference on Image Processing, 1997.

Rui, Y., Huang, T., e Mehrotra, S. "MARS and its applications to MPEG", URL: citeseer.ist.psu.edu/rui97mars.html, 1997

Sawrie, S.M., Martin, R.C., Gilliam, F.G., Faught, R.E., Maton, B., Hugg, J.W., Bush, N., Sinclair, K., Kuzniecky, R.I. "Visual confrontation naming and hippocampal function". In: Brain; 123: 770-780, 2000.

Sciascio, E. Di., Mingolla, G. e Mongiello, M. "Content-Based Image Retrieval over the Web Using Query by Sketch and relevance feedback". Proceedings of the Third International

Conference on Visual Information and Information Systems (VISUAL'99), Lecture Notes in Computer Science, v. 1614, 123-130, 1999.

Searls, D.B. "The computational linguistics of biological scientists in Artificial Intelligence and Molecular Biology", In: AAAI Press, L. Henter, 47-120, 1993.

Serapião, A. B. S.. Sensor: An Artificial Visual System. Progress Neurobiology, v. 53, no.2, pp. 179-183, 1997.

Serapião, A. B. S. "Sensor: Um Sistema Sensorial Inteligente Distribuído". Tese de Mestrado, Universidade Estadual de Campinas – Faculdade de Engenharia Elétrica – Departamento de Computação e Automação. 1996.

Serapião, A. B. S., Rocha, A. F., Rebello, M. P. F. Toward a Theory of Genetic Computing. In Genetic Algorithms and Soft Computing, F. Herrera and J. L. Verdegay, (Eds) (Physica Verlag, Heidelberg); pp. 68-94, 1996.

Shihari, R. K., Zhang, Z. e Rao, A., "Intelligent indexing and semantic retrieval of multimodal documents. Information Retrieval, 2(2):245-275, 2000.

Sikora, T. The MPEG-7 visual standard for content description - an overview, IEEE Trans. Circuits Systems Video Technol., 11, pp 702-796, 2001.

Sipser, M. Introduction to the Theory of Computation. PWS Publishing Company, Boston, 1997.

Smith, E.E., Jonides, J., Marshuetz, C., Koeppel, R. "Components of verbal working memory: Evidence from neuroimaging". In: Proceedings of National Academy Science. 95: 876-882, 1998.

Smith, J.R. e Chang, S.F. "Visualseek: A fully automated content-based image query system. Proceedings of ACM Multimedia Conference, 87-89, Boston, USA, 1996.

Souza, J. A. “Reconhecimento de Padrões Usando Indexação Recursiva”. Tese de Doutorado, Universidade Federal de Santa Catarina – Departamento De Engenharia De Produção E Sistemas. 1999.

St George, M., Kutas, M., Martines, A., Sereno, M.I. “Semantic integration in reading: engagement of the right hemisphere during discourse processing”. In: *Brain*. 122: 1317-1325, 1999.

Takahashi, N., Iwasaki, M., Kunieda, T., Wakita, Y., Day, N. Image retrieval using spatial intensity features, *Signal Processing: Image Communication* 16, pp 45-57, 2000.

Thompson, P.M., Diehd, J.N., Woods, R.P., MacDonald, D., Evans, A.C., Toga, A.W. “Growth patterns in the developing brain detected by using continuum mechanical tensor maps”. In: *Nature*; 404: 190-193, 2000.

Ungerleider, L.G., Coutney, S.M., Haxby, J.V. “A neural system for human visual working memory”. In: *Proceedings of National Academy Science*. 95: 883-890, 1998.

Van House, N.A., Butler, M. H., Ogle V. E, e Schiff, L. “User-Centered iterative Design for Digital Libraries. The Cypress experience.” *D-Lib Magazine*, 1996.

Veltkamp, R.C. e Tanase M. “Content-Based image retrieval systems: A survey.” Technical Report UU-CS-2000-34, Department of Computing Science, Utrecht University, 2000.

Wang, J. Z., LI, J., Wiederhold. SIMPLIcity : Semantics-sensitive integrated matching for picture libraries, *IEEE Trans. PAMI* 23 pp 1–17, 2001.

West, W.C., Holcomb, P.J. “Imaginal, semantic, and surface-level processing of concrete and abstract words: An electrophysiological investigation”. In: *Journal of Cognitive Neuroscience*.12: 1024-1037, 2000.

Young, M.P. “Open questions about the neural mechanisms of visual pattern recognition”. In: Gazzaniga, M.S. *The Cognitive Neuroscience*. Cambridge, The MIT Press, 463-474, 1995.

Zadeh, L.A. "A New Direction in AI – Toward a Computational Theory of Perceptions". In: AI Magazine.73-84, 2001.

Zadeh, L.A. "The linguistic approach and its application to decision analysis". In: Ho, Y.C., Mitter, S.K.. Directions in Large Scale Systems. Plenum Press, New York, 339-370, 1976.

Zadeh, L.A. "The concept of a linguistic variable and its application to approximate reasoning I, II, III". In: Information Sciences. 8: 199-251, 301-357; 9: 43-80, 1975.

Zadeh, L.A., "Fuzzy Sets". In: Information and Control.v 8(3): 338-353, 1965.

Zeki, S., Marini, L. "Three cortical stages of colour processing in the human brain". In: Brain.121: 1669-1685, 1998.

Zhou, X. S., Huang, T. S. Edge-based structural features for content-based image retrieval, Pattern Recognition Lett. 22, pp 457-468, 2001.

Zhou, X. S., Rui, Y., Huang, T. S.. Water-Filling: a novel way for image structural feature extraction, in: Proceedings of the IEEE Conference on Image Processing, Kobe, Japan, pp 570-574, 1999.

Zimmermann, H.J. Fuzzy Set Theory and Its Applications. Kluwer Academic Publishers, 1996.

Apêndice A

Sistemas de recuperação de Imagens

A seguir será apresentado um resumo de alguns sistemas de recuperação de imagens desenvolvidos dentro da área de visão computacional que foram estudados ao longo do presente trabalho e foram descrito por Souza, 1999. Foram selecionados sistemas que trabalham em diferentes domínios de aplicação, tais como: processamento da linguagem natural, detecção de faces para imagens acompanhadas de textos (em jornais), banco de dados espaciais e geográficos, coleções de arte, história, fotos, aplicações para bibliotecas e coleções de imagens distribuídas na Web.

Foram destacadas as abordagens que esses sistemas oferecem para: linguagem e interface de consulta e critérios de similaridade.

DISIMA

Em (Ozsu et al, 2000) descreve-se o sistema DISIMA. A imagem é definida em duas camadas. A primeira camada define a imagem como a unidade básica do modelo. Uma imagem é definida com quatro propriedades: o identificador; o conjunto de representações em formatos padrões de imagem (gif, jpg); o conjunto de dados alfanuméricos associados e o conteúdo definido como o conjunto dos objetos relevantes. A segunda camada captura o conteúdo das imagens. Uma imagem é vista do ponto de vista físico e lógico. Do ponto de vista físico, uma imagem é o conjunto dos objetos relevantes com seus relacionamentos espaciais. Um objeto, do ponto de vista lógico, é usado para dar semântica a um objeto físico. O modelo fornece ferramentas para manipular hierarquias de objetos (físicos e lógicos) e hierarquias definidas pelos usuários.

QBIC

O sistema QBIC (Query by Image Content) (Flickner, Sawhney e et al, 1995) (Lee, Barber, Petkovic e et al., 1994) é um sistema de recuperação de imagens baseado nas características de

cor, textura e forma. O QBIC foi desenvolvido no Centro de pesquisas Almaden IBM e seu protótipo foi escrito em linguagem de programação C, com a interface escrita em X/Motif. O sistema consiste de três partes principais:

1. formar a base de dados, onde as imagens são importadas para dentro do sistema e, opcionalmente, são acrescentados objetos esboçados e textos e palavras-chave;
2. cálculo das características, onde as características das imagens e objetos são calculadas e armazenadas na base de dados;
3. processa-se a pergunta, revisam-se os resultados e, como queira, refina-se as perguntas.

De acordo com esses passos, obtêm-se a arquitetura do sistema QBIC.

No processo de indexação do QBIC, a identificação dos objetos é o passo chave. QBIC tem devotado considerável esforço no desenvolvimento de ferramentas com objetivo de identificação de objetos. Usa métodos de segmentação não supervisionada, completamente automática para identificar objetos em uma classe restrita de imagens. Também usa ferramentas semi-automáticas para a identificação, como o método Flood-fill, que começa em um pixel do objeto simples e repetidamente adiciona pontos adjacentes cujos valores estão dentro de um dado limite do pixel original.

Após a identificação dos objetos, o QBIC calcula as características de cada objeto e imagem. No sistema QBIC, ou a cena é a imagem inteira ou um simples quadro de vídeo e um objeto é uma parte da cena. Ele utiliza conjuntos de características que capturam a cor, textura e forma das imagens. Suporta duas classes do conteúdo de cor – local e global. No QBIC a extração de regiões locais (objetos) é feita manualmente, pois requer uma pessoa para desenhar os limites da região usando um mouse. As informações da cor local e global são representadas pela média das cores e o histograma de cor. O sistema também usa uma métrica da distância quadrada para comparação dos histogramas. Como a medida da distância quadrada é muito intensa computacionalmente, a distância de cor média é usada como um pré-filtro para as perguntas de cor.

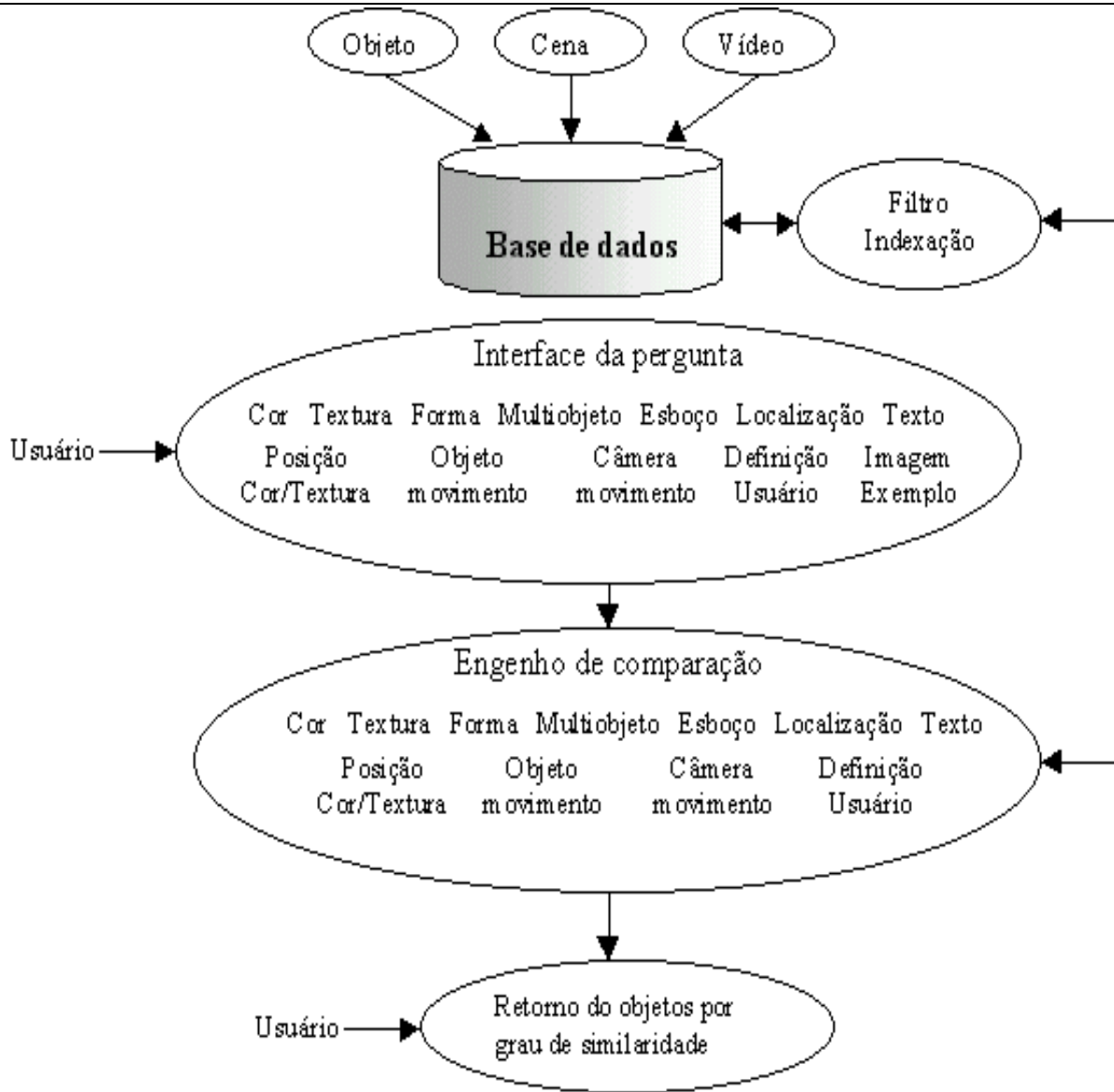


Figura A-0-1 - Arquitetura do sistema QBIC

Amore

O Amore (Mukherjea, Hirata e Hara, 1999) é um sistema baseado na WEB que considera a Internet como um grande repositório de imagens. Ele indexa os sítios WEB por textos e imagens contidas/referenciadas nos documentos HTML. As imagens são segmentadas em regiões homogêneas de cores. Nas consultas também podem especificar um URL.

Blobworld

Blobworld (Carson, Thomas e et al., 1999) é um sistema de recuperação de imagens baseado no conteúdo. As imagens são automaticamente segmentadas em regiões que correspondem aproximadamente a objetos ou partes de objetos. O usuário escolhe, a partir de uma imagem inicial, uma região e indica o grau de importância para a consulta. Em seguida, indica a importância da textura, cor, localização e forma da região. Nas consultas podem ser utilizadas mais de uma região.

VisualSeek

O VisualSeek (Smith e Chang, 1996) é um sistema híbrido que integra indexação de imagem baseada em conteúdo (cor e textura) e em propriedades espaciais (como tamanho, localização e relações com outras regiões). A integração confia na representação de regiões através de um conjunto de cores. Primeiro, um conjunto de cores fornece para o sistema de extração de regiões automatizada através da retro-projeção do conjunto de cores. Segundo, os conjuntos de cores são facilmente indexados para a recuperação de imagens com um conjunto de cores semelhante. Dessa forma, as imagens são comparadas através de suas regiões. Além disso, o sistema fornece ao usuário o controle da seleção das regiões e os parâmetros que são mais importantes na determinação da similaridade na pergunta dada.

O projeto do VisualSEEk enfatiza diversos objetivos únicos a fim de melhorar a recuperação de imagens, como: (1) extração automática de características e regiões localizadas, (2) examina informações espaciais e as características, (3) extração de características de dados compactados, (4) desenvolvimento de técnicas para indexação e recuperação rápida e (5) desenvolvimento de ferramentas altamente funcionais para o usuário.

A aplicação servidora do VisualSEEk foi desenvolvida na linguagem Java para permitir independência de plataforma de cliente e acessibilidade no World Wide Web. O sistema de VisualSEEk consiste em vários componentes: o conjunto de ferramentas do usuário, o servidor de perguntas, a imagem e servidor de vídeo, o arquivo de imagem e de vídeo, o banco de dados dos meta-dados e os arquivos de índice.

Atualmente, o sistema de VisualSEEk permite procurar imagens em uma base de 12.000 imagens coloridas. As ferramentas de VisualSEEk também estão sendo portadas a uma aplicação de busca para procurar em uma coleção de mais de um milhão de imagens e vídeos da World Wide Web.

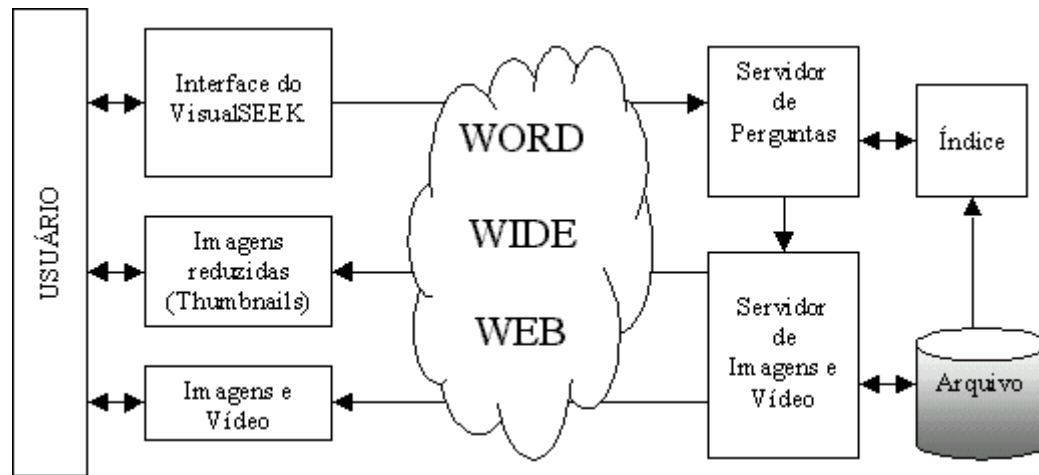


Figura A-0-2 - Arquitetura do VisualSEEK

Drawsearch

O Drawsearch (Di Sciascio, Mingolla e Mongiello, 1999) é um sistema protótipo que utiliza técnicas de retro alimentação para facilitar a interação com o usuário. DrawSearch tem duas interfaces. A primeira permite consultas baseadas em cor e forma. O usuário faz um desenho (sketch) colocando cores nas linhas e regiões fechadas. O segundo subsistema permite consultar escolhendo textura.

Cypress Chabot

O Cypress (Van House, Butler, Ogle e Schiff, 1996) é uma nova versão do CHABOT (Ogle, Stonebraker, 1995). O projeto foi definido de domínio específico com o objetivo de armazenar e recuperar imagens de uma grande coleção do *Department Water Resources – CA, USA*. A análise do conteúdo das imagens é baseada em *color blobs*, que são agrupamentos de cores dentro de uma imagem que podem corresponder a objetos, por exemplo: carro vermelho, peixe amarelo. Define consultas por conceito. Uma consulta por conceito envolve informações contextuais como “céu azul”. O conceito “céu azul” inclui imagens com a palavra “céu” no

campo de descrição e imagens com predominância da cor azul. A interface com o usuário para consultas tem acesso através da Web.

MARS - Multimedia Analysis Retrieval Systems

O projeto MARS proposto em (Rui, Huang e Mehrotra, 1997), (Rui, Huang e Mehrotra, 07-1997) e (Velkamp e Tanase, 2000) descreve as ferramentas de extração de características, o modelo de objetos e o modelo de recuperação. O modelo de objetos descreve o conteúdo multimídia (imagens) e os esquemas de indexação utilizando como base a interface MPEG-7. O modelo inclui vários descritores para cada característica (cor, textura, forma, layout, etc.), para suportar a subjetividade dos usuários.

MIR

MIR – Multimodal Information Retrieval System (Shihari, Zhang, e Rao, 2000) combina várias técnicas de processamento de textos e de imagens para induzir descrições semânticas das imagens. Utiliza o processamento da linguagem natural para extrair descrições semânticas como: localizações, relacionamentos espaciais entre pessoas e características que ajudem no reconhecimento de faces. Faz um reconhecimento de faces em um modelo de três contornos (linha do cabelo, contornos esquerdo e direito).

NETRA

Em NETRA (Ma e Manjunath, 1999) as imagens são automaticamente segmentadas em regiões de cor homogênea. Para cada região são extraídas a cor, textura, forma e relacionamentos espaciais. As consultas podem ser feitas utilizando exemplos de imagens ou especificando cores ou relacionamentos.

SEMCOG

(Li, Candan, Hirata, Hara, 1997) e (Hirata, Hara, Li e Candan, 1997) tentam combinar a linguagem natural do usuário a uma linguagem precisa do computador. O usuário, de maneira visual, constrói a consulta que é traduzida para a linguagem CSQL (Cognition-SQL). Em uma consulta é possível definir um peso aos objetos para definir a importância dos predicados, e é

possível pesquisar atributos associados às imagens. As consultas são principalmente baseadas nos objetos das imagens, e não consideram a imagem tratada como única entidade.

CHITRA

No sistema CHITRA (Nepal e Ramakrishna, 1999) (Ramakrishna e Nepal, 2000) permite definir conceitos abstratos (como sunset) em termos das características das imagens (como textura e cor). Foi definida uma extensão fuzzy da linguagem de consulta (Nepal, Ramakrishna e Thom, 1999) a fim de permitir obter resultados com uma determinada relevância para a consulta, assim como permite especificar o número de imagens relevantes a serem recuperadas. A definição de tipos de dados fuzzy captura melhor as comparações entre as imagens baseadas em similaridade, mas o sistema é endereçado à solução do processamento e otimização de consulta e não fornece ferramentas para a modelagem, representação e apresentação visual das imagens.

MARS – Multimedia Authoring Retrieval Service

O sistema MARS (Henrich e Robbert, 2000) desenvolveu uma extensão à linguagem declarativa de consulta (OQL-ODMG) (Cattell, Barry. e et al.,2000) chamada de POQL^{MM}, uma linguagem de propósito geral orientada a documentos multimídia estruturados. Foi desenvolvida uma interface visual que traduz a consulta a POQL^{MM}. A interface apresenta três seções: seleção por atributos das imagens, onde pode ser representado um esquema (sketch) ou um exemplo (QBE – Query by Example), considerando relacionamentos espaciais, histograma de cores, textura e forma. As expressões acabam sendo complexas e a interface visual possui elementos que podem não ser compreensível ao usuário final.

A seguir é apresentada uma tabela com os sistemas descritos acima.

Sistema	Descritores	Similaridade
QBIC	Cor: vetor de 3 dim. da média das cores no espaço Munsell.	Distância euclidiana ponderada.
	Histograma de cores no espaço RGB com vetor de dim. 256	Medida quadrática
	Textura: vetor de 3 dim	Distância euclidiana ponderada.
	Forma: vetor com elementos: área, circularidade, orientação dos eixos, etc.	Distância euclidiana ponderada.
	Relacionamentos espaciais: localização do centro do objeto na imagem	Segmentação em regiões, cada região é espacialmente analisada
	Sketch: mapa de baixa resolução 64x64	
	Palavras chaves (separada)	
Amore	Cor, Forma: Segmentação em 8 regiões por cores homogêneas	Região a Região: Forma: Pixels sobrepostos Cor: distância no espaço HLS
	Palavras Chaves	
Blobworld	Cor: histograma de 218 dim. no espaço por região	Distância quadrática
	Textura: (contraste, anisotropy) por região	Distância euclidiana
	Forma: (área, centro, orientação)	Distância euclidiana entre os centros
	Combinação: as distâncias são combinadas em uma distância final	
VisualSeek	Textura: vetor de 9 dim	Combinação: Matriz para textura e cor, a partir da qual é computada uma distância.
	Cor: segmentação e histograma de cores	
	Forma: MBR, área e centro de regiões	Distância Euclidiana
	Combinação: soma das distâncias ponderadas dá uma distância combinadas	
Drawsearch	Cor: imagem dividida em 4x4. Para cada subimagem: média de cores em 48 dim.	Cor e forma: métrica do coseno para os vetores
	Forma: os elementos do vetor descritor são a parte real dos 100 coeficientes de <i>Fourier</i>	Combinação: soma ponderada das distâncias entre o vetor de cor e o descritor da forma
	Textura: segmentação usando <i>Gaussian Markovian Random Fields</i>	Distância euclidiana
Cypress Chabot	Cor: busca por <i>color blobs</i> , correspondência com objetos (carro vermelho) Consultas por conceito	Cor: métrica de distância Combinação: textos & <i>color blobs</i>

Sistema	Descritores	Similaridade
MARS (<i>Multimedia Analysis and Retrieval Systems</i>)	Cor: histograma sobre espaço HSV	Interseção de histogramas
	Textura: vetor de 3 dim. CCD	Distância euclidiana
	Esquema (<i>Layout</i>): 5x5 subimagem e p/ cada uma, histograma HS de 2 dim. (cor, textura)	Distância entre duas imagens: soma das 5x5 similaridades
	Forma: MFD	Distância euclidiana
	Palavras chaves	
MIR	Processamento da Linguagem Natural para extrair descrições semânticas como: localizações, relacionamentos espaciais entre pessoas e características que ajudem no reconhecimento de faces. Reconhecimento de faces: modelo de 3 contornos (linha do cabelo, contornos esquerdo e direito).	Combinação: Soma parametrizada das similaridades dos diferentes processamentos (textos e CIs)
	Cor: histograma	Distância euclidiana
NETRA	Segmentação de imagens em regiões de cores homogêneas. Para cada região:	
	Cor: vetor de n dim. (n = número de cores para representar a região)	Distância euclidiana no espaço RGB
	Forma: FFT de 3 vetores usados para Representar a forma das regiões	<i>L1-distance</i>
	Textura: vetor que representa Transformada de <i>Gabor</i>	Distância euclidiana
DISIMA	Objetos e relacionamentos espaciais: □ detecção de faces usando o <i>Minimum Bounding Rectangle</i> MBR. Palavras chaves: anotações sobre as faces.	Predicado <i>contains</i> (objeto contido em imagem). Distâncias de cor e textura sobre imagens. Distâncias sobre objetos. Buscas tradicionais sobre textos. Combinações de todos.

Sistema	Descritores	Similaridade
SEMCOG	Objetos e relacionamentos espaciais: Sistema COIR: identifica regiões (objetos)	COIR: identifica relacionamentos espaciais dos objetos e
	Catálogo: semântica s/ componentes das imagens	semântica usando o catálogo
CHITRA	Cor: histograma de 768 elementos	Algum critério de similaridade para cada CI.
	Textura: vetor de dim. 48 com funções de <i>Gabor</i>	
MARS (<i>Multimedia Authoring Retrieval Service</i>)	QBE: cor (histograma), textura e forma. <i>Sketch</i> : representando relacionamentos Espaciais. Atributos textuais e outros tipos de mídia.	Cor, forma: distância euclidiana. Combinação: soma das distâncias ponderadas.

No quadro resumo acima são apresentadas as propriedades dos sistemas de reconhecimento de imagens discutidos acima. As colunas do quadro acima fazem uma classificação baseada nos seguintes pontos:

Descritores: Características visuais da imagem de baixo ou alto nível e seu processamento para extrair a informação;

Similaridade: os critérios de similaridade para cada descritor, utilizados para comparar a imagem da consulta (ou esboço, textura, cor, etc.) com as imagens no banco;

As consultas em cada sistema são baseadas nos descritores e as comparações são feitas utilizando as operações de similaridade correspondente. No caso de atributos tradicionais e as palavras chaves, as consultas são feitas da maneira tradicional.

Mais informações sobre sistemas de reconhecimento de imagens já desenvolvidos podem ser obtidos em Veltkamp e Tanase (2000), que faz uma revisão de trinta e nove deles. Cada sistema é descrito baseado em características, apresentação, consulta, indexação, aplicação, critério de similaridade.

Algoritmos

Os algoritmos que utilizam essas aplicações podem ser classificados em três tipos fundamentais:

1. *algoritmos de extração de descritores*: são as operações de segmentação e de extração de descritores das imagens;
2. *algoritmos de comparação ou similaridade*: são as operações que comparam as características ou informações semânticas de duas ou mais imagens;
3. *algoritmos de indexação*: a cada característica da imagem pode ter associado um método de acesso (por questões de otimização).

Arquiteturas

A maioria das arquiteturas aqui apresentadas é composta por duas fases. A primeira fase refere-se à etapa de pré-processamento das imagens, onde ocorre o povoamento dos bancos de dados. Os descritores de uma imagem são extraídos e armazenados em um banco de dados, e são criadas as estruturas de indexação.

Na segunda etapa, o usuário final de uma aplicação monta uma consulta que vai ser processada. Uma vez montada, a consulta é processada. No caso da consulta por esboço/imagem, os descritores apropriados são extraídos e comparados aos armazenados no banco de dados e assim as imagens similares são recuperadas e os resultados são visualizados.

Primitivas de Consultas sobre Imagens

A estrutura complexa das imagens e sua semântica são de difícil representação pelas linguagens de consultas utilizadas em bancos de dados tradicionais, por exemplo, SQL. A propriedade visual inerente às imagens necessita de meios para que o usuário possa consultar o banco de dados utilizando representações visuais.

A maioria dos sistemas suportam uma ou mais das primitivas. As consultas sobre imagens podem ser feitas como uma combinação das seguintes primitivas (Aslandogan e Yu, 1999):

-
- consultas seqüenciais: O usuário percorre seqüencialmente todas as imagens do banco de dados;
 - consultas por conceitos: O usuário descreve diretamente com palavras chave ou na linguagem natural as características semânticas do conteúdo das imagens buscadas. Os conceitos são baseados nos descritores. Por exemplo: um conceito *céu azul* pode ser definido como aquelas imagens com a *cor* azul predominante na metade superior;
 - consultas por descritores: O usuário descreve diretamente as características visuais do conteúdo das imagens buscadas (por exemplo: cor, textura e forma);
 - consultas por atributos tradicionais: Os usuários especificam os atributos não visuais das imagens buscadas, tais como atributos físicos (localização, armazenamento: tamanho, compressão), biográficos (autor, data, título) ou outros associados (subtítulos, gravações);
 - consulta por esboço ou imagem: O usuário oferece uma imagem como amostra ou desenha seus requisitos. A similaridade pode ser determinada baseada na característica escolhida pelo usuário;
 - consultas por categorias: A interface de consulta é organizada por categorias. O usuário navega pelas diferentes categorias ou escolhe uma a partir da qual expressa a consulta utilizando uma das classes anteriores.

Apêndice B

Descrição das funções implementadas para filtros de Imagem

FilterReverse

Inverte as cores de todos os pontos da imagem.

FilterMedian

Aplica o filtro “Median” usando uma área de 3x3. Os 9 pontos são ordenados e pega-se o valor do meio da lista como sendo o novo valor do pixel.

FilterGray

A imagem é transformada em tons de cinza de acordo com a fórmula:

$$0.3 * R + 0.59 * G + 0.11 * B$$

Filter3x3

Aplica um filtro 3x3 por convolução.

m11	m12	m13
m21	m22	m23
m31	m32	m33

/ divisor

Se o novo valor for menor que zero, usa o valor zero para a imagem.

Se o novo valor for maior que 255, usa o valor 255 para a imagem.

O filtro é aplicado nas 3 cores da imagem independentemente.

FilterBlur

Aplica o filtro 3x3.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

FilterHistogramEqualizer

Faz a equalização do histograma.

Calcula o Histograma Acumulado dentro da função (afeta a função Histograma).

FilterBW

Transforma a imagem em uma imagem de 2 cores, recebe dois parâmetros: o valor de threshold 1 e 2.

0 se cor < threshold1 e 1 se cor > threshold2

255 se threshold1 < cor < threshold 2

Na chamada, caso threshold2 seja menor que threshold1, utiliza 255 em threshold2.

FilterContour

Filtro para extração do contorno mais preciso que o usado tradicionalmente.

Aplica um filtro 2x2

0	-1
-1	2

E obtêm o módulo (valor absoluto) do resultado ao invés de “cortar” valores negativos

FilterAreasMagic

Faz uma rotulação inteligente de áreas sem a necessidade de outros filtros.

Rotula uma área utilizando o algoritmo de preenchimento inteligente descrito no Apêndice C.

Recebe como parâmetros

- o tamanho mínimo de área aceita (MinSize) – áreas com tamanho menor que este são descartadas.

- o limite de variação aceita na cor (LimiteCor)

- o limite de proximidade da borda da imagem – áreas com proximidade menor ou igual a esta são automaticamente descartadas, independente de seu tamanho

FilterAreasMagic2

Além dos parâmetros de tamanho mínimo de área aceita e do limite de variação na cor recebe as coordenadas onde será aplicado (xmin, ymin, xmax e ymax).

Filtros utilizados durante o desenvolvimento do sistema

Foram utilizadas 5 configurações de filtros diferentes, sendo duas objetivando a extração de silhuetas da imagem, e os outros 3 com o objetivo de extrair os componentes da imagem com parâmetros diferentes para cada um desses 3 filtros.

Filtro 1 – Extração da Silhueta

Este filtro tem como objetivo extrair a silhueta das figuras contidas na imagem, descartando os objetos que estão em contato com a borda.

A macro deste filtro é:

```
C.CopyPic  
C.FilterAreasMagic 30, 80, 2  
C.FilterBlur  
C.FilterGray  
C.FilterBW 10, 0  
C.FilterAreasMagic 30, 30, 1
```

A figura é armazenada para que as cores da silhueta possam ser calculadas, então é feita a rotulação excluindo as áreas que atingem uma borda de 2 pontos da figura. Após a rotulação é feita a expansão da figura para unir as áreas através da seqüência Blur, Gray e BW. Após isso é feita uma nova rotulação.

Filtro 2 – Extração da Silhueta

Este filtro tem com objetivo extrair áreas da imagem (rotulação) utilizando o filtro FilterAreasMagic com um fator de semelhança de 30 e excluindo áreas com área menor que 30 pontos, também excluindo as áreas que encostam-se às bordas da imagem.

A macro deste filtro é:

```
C.FilterAreasMagic 30, 30, 1
```

Filtro 3 – Extração da Silhueta

Idêntico ao Filtro 2, porém com parâmetro de semelhança de cor de 50 e excluindo áreas menores que 80 pontos.

A macro deste filtro é:

```
C.FilterAreasMagic 80, 50, 1
```

Filtro 4 – Extração da Silhueta

Este filtro tem o mesmo objetivo do Filtro 2, porém são aplicados os filtros Median e Blur a fim de excluir ruídos na imagem.

A macro deste filtro é:

```
C.FilterMedian
```

```
C.FilterBlur  
C.FilterAreasMagic 70, 60, 1
```

Filtro 5 – Extração da Silhueta

Este filtro é utilizado para a extração de silhuetas de uma área específica da imagem. Ele é utilizado para a comparação do formato da silhueta em áreas da imagem cujo limite foi definido por características já encontradas na imagem, como tamanho, cor e proximidade de áreas.

A macro deste filtro é:

```
C.SetLimits @  
C.FilterAreasMagic 30, 80, 2  
C.FilterBlur  
C.FilterGray  
C.FilterBW 10, 0  
C.FilterAreasMagic 30, 30, 3
```

O parâmetro @ representa os valores Xmin, Ymin, Xmax e Ymax da área onde a silhueta deve ser extraída.

Ele é semelhante ao Filtro 1, com a diferença de que a área de atuação do filtro dentro da figura é definida, e de que a figura inicial não necessita ser armazenada, pois não é necessário obter a cor média das áreas encontradas.

Extração de áreas

Contorno

NumAreas();

Retorna o número de áreas encontradas na imagem.

AreaID(long x, long y);

Devolve o número da área a que pertence o ponto (x, y).

Devolve zero caso (x, y) não pertença a nenhuma área ou as áreas não tenham sido definidas.

As funções para definição das áreas são

- FilterAreasMagic
- FilterAreasMagic2

AreaNum(short AreaNum);

Devolve uma cadeia de caracteres com a descrição da área informada no parâmetro.

Devolve uma cadeia vazia caso a área não exista.

A cadeia retornada segue o formato:

Cor#Tamanho#Significativos#Cadeia#(Xini,Yini)#(Xcentro,Ycentro)#Pontos

onde:

Cor = inteiro longo com a cor da área imagem (RGB codificado em inteiro longo)

Tamanho = Tamanho da área extraída

Significativos = lista dos pontos significativos encontrados na área no formato X; Y; P/ onde P = pertinência do ponto (distância com relação à semi-reta entre o ponto anterior e o ponto posterior)

Cadeia = É a assinatura da área. Ela se encontra no formato pDa , onde:

p = pertinência (de 'a' a 'h')

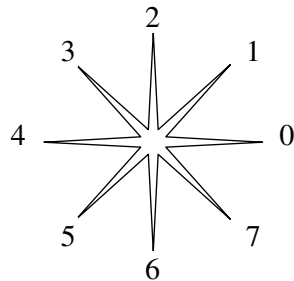
D = distância (de 'A' a 'H')

a = ângulo (de 0 a 7) – ângulo relativo ao ângulo anterior

(Xini, Yini) = Pontos iniciais da área em relação a figura, o ponto mais baixo a esquerda.

(Xcentro, Ycentro) = centro geométrico da área

pontos = direções dos pontos que compõem a borda da área, segundo a orientação mostrada figura abaixo.



Orientação dos pontos

Abaixo temos um exemplo de uma cadeia de caracteres retornada pelo sistema de certa área:

13619199#39#473;24;36/473;36;36/#bB4bB4#(472,25)#(473,31)#107666666665534322222221

Cor: 13619199	Tamanho: 39
Pontos: 473;24;36/473;36;36/	Cadeia: bB4bB4
(Xini, Yini): (472, 25)	(Xcentro, Ycentro): (473, 31)
Curva: 107666666666553432222221	

Apêndice C

Rotulação Inteligente

Para o preenchimento de áreas para a rotulação foi usado o algoritmo FloodFill com alterações, pois o algoritmo original recursivo não é aplicável a grandes áreas pois causa estouro de pilha.

```
void floodFill(xstart, ystart, oldColor, newColor)
{
    if(color(xstart, ystart) == oldColor){
        setColor(xstart, ystart, newColor);
        boundaryFill(xstart+1, y, 0, oldColor, newColor);
        boundaryFill(xstart, y+1, 0, oldColor, newColor);
        boundaryFill(xstart-1, y, 0, oldColor, newColor);
        boundaryFill(xstart, y-1, 0, oldColor, newColor);
    }
}
```

Foi retirada grande parte da recursividade do algoritmo.

```
void BoundaryFill(long x, long y, int direcao, long corVelha, long
corNova)
{
    LPBYTE p;
    long p1, p2;

    if (getColor(x, y) == corVelha)
    {
// Loop para y
        do{
            p1 = -1;
            p2 = -1;

            if ((direcao == 0) && (x > 0)) BoundaryFill(x-
1, y, 1, corVelha, corNova);

// Loop para x
            do{
                putColor(x, y, corNova);

                if (y < Altura){
                    if (getColor(x, y+1) == corVelha){
                        p1 = x;
                    }else{
```

```

        if (p1>=0){
            BoundaryFill(p1,y+1, 1 -
direcao,corVelha,corNova);
            p1=-1;
        }
    }

    if (y>0){
        if (getColor(x,y-1)==corVelha){
            p2=x;
        } else {
            if (p2>=0){
                BoundaryFill(p2,y-1, 1 -
direcao,corVelha,corNova);
                p2=-1;
            }
        }
    }

    if (direcao==0){
        if (x<Largura) x++;
    } else{
        if (x>0) x--;
    }
    // Se não alterar x, getColor(x,y) será igual a corNova
    } while (getColor(x,y)==corVelha);

    if (p2>=0)BoundaryFill(p2,y-1,
0,corVelha,corNova);

// Próxima linha de forma não recursiva
    if (p1>=0){
        x = p1;
        y++;
        direcao = 0;
    }
    } while (p1>=0);
}
}

```

Filtro usado para rotação inteligente das áreas

```

#define compara_cores(R1,G1,B1,R2,G2,B2,l) ( (abs((int)R1-R2)<l)
&& (abs((int)G1-G2)<l) && (abs((int)B1-B2)<l))

```

```

void CCurvaCtrlCtrl::MarcaArea2(long x,long y, int direcao){
    long p1,p2;

```



```
LPBYTE p,pa;
byte cor_R,cor_G,cor_B;

p1=-1;
p2=-1;

if ((x<0) || (y<0) || (x>=m_Width) || (y>=m_Height)) return;
if (lpBMPArea[y*m_Width+x]!=0) return;

p=pixel(x,y);
if (size_area>0){
    cor_R= sum_corR / size_area;
    cor_G= sum_corG / size_area;
    cor_B= sum_corB / size_area;
}else{
    cor_R=p[0];
    cor_G=p[1];
    cor_B=p[2];
}

if
(compara_cores(cor_R,cor_G,cor_B,p[0],p[1],p[2],cor_limite)){
    if (direcao==0){
        if (x>0) {
            pa=pixel(x-1,y);
            if ((lpBMPArea[y*m_Width+x-1]==0) &&
compara_cores(cor_R,cor_G,cor_B,pa[0],pa[1],pa[2],cor_limite))
                MarcaArea2(x-1,y,1);
        }
    }

    while ((lpBMPArea[y*m_Width+x]==0) &&
compara_cores(cor_R,cor_G,cor_B,p[0],p[1],p[2],cor_limite)){
        lpBMPArea[y*m_Width+x]=label_count;
        if (AreaXmin>x) AreaXmin=x;
        if (AreaXmax<x) AreaXmax=x;
        if (AreaYmin>y) AreaYmin=y;
        if (AreaYmax<y) AreaYmax=y;
        size_area++;
        sum_corR+=p[0];
        sum_corG+=p[1];
        sum_corB+=p[2];
        cor_R= sum_corR / size_area;
        cor_G= sum_corG / size_area;
        cor_B= sum_corB / size_area;
```

```

        if (y<m_Height-1){
            pa=pixel(x,y+1);
            if ((lpBMPArea[(y+1)*m_Width+x]==0) &&
compara_cores(cor_R,cor_G,cor_B,pa[0],pa[1],pa[2],cor_limite)){
                p1=x;
            }else{
                if (p1>=0){
                    MarcaArea2(p1,y+1,0);
                    p1=-1;
                }
            }
        }

        if (y>0){
            pa=pixel(x,y-1);
            if ((lpBMPArea[(y-1)*m_Width+x]==0) &&
compara_cores(cor_R,cor_G,cor_B,pa[0],pa[1],pa[2],cor_limite)){
                p2=x;
            }else{
                if (p2>=0){
                    MarcaArea2(p2,y-1,0);
                    p2=-1;
                }
            }
        }

        if (direcao==0){
            if (x<m_Width-1) x++;
        } else{
            if (x>0) x--;
        }

        p=pixel(x,y);
    }

    if (p1>=0)    MarcaArea2(p1,y+1,0);
    if (p2>=0)    MarcaArea2(p2,y-1,0);
}
}

```