

EVERTON VIDAL VIEIRA

**MINERAÇÃO DE IMAGENS: CONCEITOS E APLICAÇÃO EM SISTEMAS
DE RECUPERAÇÃO DE IMAGENS POR CONTEÚDO**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientadora: Prof.^a Dr.^a Olga R. P. Bellon

CURITIBA

2002



Ministério da Educação
Universidade Federal do Paraná
Mestrado em Informática

PARECER

Nós, abaixo assinados, membros da Banca Examinadora da defesa de Dissertação de Mestrado em Informática, do aluno *Everton Vidal Vieira*, avaliamos o trabalho intitulado, "*Mineração de Imagens: Conceitos e Aplicações em Sistemas de Recuperação de Imagens por Conteúdo*", cuja defesa foi realizada no dia 16 de outubro de 2002, às nove horas, no anfiteatro A do Setor de Ciências Exatas da Universidade Federal do Paraná. Após a avaliação, decidimos pela aprovação do candidato.

Curitiba, 16 de outubro de 2002.

Prof.^a. Dra. Olga Regina Pereira Bellon
DINF/UFPR

Prof.^a. Dra. Agma Juci Machado Traina
HC/ICMC/USP

Prof. Dr. Alexandre Ibrahim Direne
DINF/UFPR

AGRADECIMENTOS

Gostaria de agradecer aos meus pais e irmã pelo apoio, incentivo e paciência durante o período de elaboração deste trabalho. Agradeço também ao CITS (Centro Internacional de Tecnologia de Software) pela liberdade que me foi concedida de poder me dedicar ao Mestrado. Destaco também a colaboração de amigos, colegas de curso e de trabalho.

Também sou grato aos pesquisadores do grupo Imago, em especial à professora Olga Regina Pereira Bellon pela atenção e orientação, como também ao Luciano Silva, pela ajuda na execução dos experimentos.

SUMÁRIO

LISTA DE ILUSTRAÇÕES	viii
LISTA DE TABELAS	ix
RESUMO	x
ABSTRACT	xi
1 INTRODUÇÃO.....	1
1.1 OBJETIVOS.....	3
1.2 ORGANIZAÇÃO.....	3
2 CONCEITOS.....	5
2.1 MINERAÇÃO DE DADOS.....	5
2.1.1 Banco de Dados.....	7
2.1.2 Aprendizado de Máquinas.....	7
2.2 VISÃO COMPUTACIONAL.....	8
2.3 SISTEMAS DE RECUPERAÇÃO DE IMAGENS POR CONTEÚDO.....	9
2.4 DEFINIÇÃO DE MINERAÇÃO DE IMAGENS.....	11
3 PROCESSO DE MINERAÇÃO DE IMAGENS.....	13
3.1 OBTENÇÃO.....	13
3.2 PRÉ-PROCESSAMENTO.....	14
3.3 EXTRAÇÃO DE CARACTERÍSTICAS.....	16
3.4 INTEGRAÇÃO.....	17
3.5 MODELAGEM DO BANCO DE DADOS.....	17
3.5.1 Bancos de dados relacionais e orientados a objeto.....	19
3.5.2 Bancos de dados multidimensionais.....	20
3.5.3 Bancos de dados semi-estruturados.....	20
3.5.4 Bancos de dados multimídia e imagens.....	21
3.5.5 Bancos de dados espaciais.....	21
3.6 INDEXAÇÃO.....	22
3.7 MINERAÇÃO.....	23
3.8 REFINAMENTO.....	23
3.9 APLICAÇÃO E TESTE.....	23

4	ESTRATÉGIAS DE MINERAÇÃO DE DADOS E IMAGENS.....	25
4.1	CLASSIFICAÇÃO	25
4.2	DESCOBERTA DE REGRAS DE ASSOCIAÇÃO.....	26
4.3	AGRUPAMENTO	26
4.4	OUTRAS TAREFAS.....	27
5	ALGORITMOS E APLICAÇÕES.....	28
5.1	ÁRVORES DE DECISÃO E INDUÇÃO DE REGRAS.....	28
5.2	ALGORITMOS DE CLUSTERIZAÇÃO	29
5.3	DESCOBERTA DE REGRAS DE ASSOCIAÇÃO.....	29
5.4	LÓGICA FUZZY	31
5.5	MÉTODO DE BAYES E REDES BAYESIANAS	31
5.6	REGRESSÃO	33
5.7	REDES NEURAIS.....	33
5.8	INSTANCE-BASED LEARNING RACIOCÍNIO BASEADO EM CASOS	34
5.9	PROGRAMAÇÃO DE LÓGICA INDUTIVA	34
5.10	COMPUTAÇÃO EVOLUCIONÁRIA E ALGORITMOS GENÉTICOS (AG).....	35
5.11	ROUGH SETS.....	36
5.12	OUTROS ALGORITMOS.....	36
6	ARQUITETURA DE UM SISTEMA DE MINERAÇÃO DE IMAGENS.....	37
6.1	CLASSIFICAÇÃO QUANTO À ABORDAGEM: ORIENTADA A MODELO X ORIENTADA A DADOS.....	37
6.2	CLASSIFICAÇÃO QUANTO À MODULARIDADE: ORIENTADO À FUNÇÃO X INFORMAÇÃO	39
6.3	CLASSIFICAÇÃO QUANTO AO TIPO DE APLICAÇÃO	40
7	PROJETO SRIC3D.....	42
7.1	TRABALHOS RELACIONADOS	42
7.1.1	Imagens de profundidade.....	42
7.1.2	Modelos tridimensionais.....	43
7.2	ARQUITETURA DO SISTEMA	44
8	FERRAMENTA DE MINERAÇÃO DE IMAGENS	50
8.1	VISÃO ARQUITETURAL	50
8.1.1	Módulo de Mineração de Dados.....	52

8.1.2	Módulo de Mineração de Imagens	54
8.1.3	SRIC	57
8.1.4	Módulo de Extração de Características	58
8.1.5	Apresentação e interação	58
8.2	DETALHAMENTO	58
8.2.1	<i>ImageSet</i>	59
8.2.2	<i>ImageInstance</i>	60
8.2.3	<i>Feature</i>	60
8.2.4	<i>Concept</i>	61
8.2.5	<i>Evaluator</i>	62
8.2.6	<i>ImageMiner</i> , <i>DistributionClassifierMiner</i> e <i>DistributionClustererMiner</i>	63
8.2.7	<i>CBIR</i> e <i>CBIRResult</i>	64
8.3	MODELO DE IMPLEMENTAÇÃO	66
8.3.1	Visão estática	66
8.3.2	Visão Dinâmica	68
8.4	IMPLEMENTAÇÃO E EXPERIMENTOS	70
8.4.1	Configuração da base de dados	70
8.4.2	Configuração dos algoritmos de mineração	73
8.4.3	Execução de consultas	75
8.4.4	Resultados experimentais	77
9	CONCLUSÃO E TRABALHOS FUTUROS	86
9.1	CONCLUSÃO	86
9.2	TRABALHOS FUTUROS	88
10	REFERÊNCIAS BIBLIOGRÁFICAS	92

LISTA DE ILUSTRAÇÕES

FIGURA 1 - INTEGRAÇÃO DA MINERAÇÃO DE IMAGENS COM OUTRAS TECNOLOGIAS	6
FIGURA 2 - PROCESSO DE MINERAÇÃO DE IMAGENS.....	15
FIGURA 3 - PRINCIPAIS ESTRUTURAS DE BANCO DE DADOS	19
FIGURA 4 - ORIENTADO A DADOS X ORIENTADO A MODELO.....	38
FIGURA 5 - CLASSIFICAÇÃO QUANTO À MODULARIDADE.....	39
FIGURA 6- ARQUITETURA DO SRIC3D	45
FIGURA 7- ATIVIDADE DE EXTRAÇÃO DE CARACTERÍSTICAS	48
FIGURA 8 - ARQUITETURA DA FERRAMENTA DE MINERAÇÃO DE IMAGENS .	51
FIGURA 9 – PRINCIPAIS CLASSES, ATRIBUTOS E MÉTODOS DO MÓDULO DE MINERAÇÃO DE IMAGENS E SRIC	55
FIGURA 10 – DIAGRAMA DE SEQÜÊNCIA DO MÓDULO DE CBIR	67
FIGURA 11 – DIAGRAMA DE COLABORAÇÃO DO MÓDULO DE CBIR.....	68
FIGURA 12 – CONFIGURAÇÃO DO BANCO DE DADOS	71
FIGURA 13 – MODELO DE DADOS.....	71
FIGURA 14– CONFIGURAÇÃO DOS ALGORITMOS DE MINERAÇÃO.....	74
FIGURA 15 – ESPECIFICAÇÃO E VISUALIZAÇÃO DE CONSULTA	75

LISTA DE TABELAS

TABELA 1: EXEMPLO DE CONCEITOS POR IMAGEM.....	56
TABELA 2: OBJETOS ARMAZENADOS NO BANCO DE IMAGENS.....	73
TABELA 3: MINERAÇÃO DE IMAGENS COM O ALGORITMO PART.....	78
TABELA 4: MINERAÇÃO DE IMAGENS COM REDES NEURAIIS.....	79
TABELA 5: MINERAÇÃO DE IMAGENS COM O ALGORITMO IBk.....	81
TABELA 6: MINERAÇÃO DE IMAGENS COM O ALGORITMO EM.....	82
TABELA 7: MINERAÇÃO DE IMAGENS COM O ALGORITMO K*.....	83
TABELA 8: SRIC COMBINANDO 5 ALGORITMOS DE MINERAÇÃO.....	84

RESUMO

A crescente facilidade de obtenção, geração e armazenamento de imagens abre a possibilidade para que novas aplicações sejam criadas, envolvendo principalmente a consulta, recuperação, análise e descoberta de conhecimento em grandes bases de imagens. Para extrair conhecimento dessa base, são necessários sistemas e algoritmos que possam ser aplicados a um grande volume de dados e ao mesmo tempo possam classificar, encontrar padrões e relacionamentos de maneira eficiente. Nesse contexto, uma nova tecnologia emerge, Mineração de Imagens, combinando técnicas de diversas áreas ou domínios. Esse trabalho tem dois objetivos principais: (1) apresentar os conceitos de Mineração de Imagens e seu relacionamento com outras disciplinas, e; (2) modelar e implementar uma ferramenta de Mineração de Imagens para ser aplicada a uma base de imagens de profundidade. Para atingir o primeiro objetivo, será apresentada uma revisão de algumas definições de Mineração de Dados e Visão Computacional, bem como as fases que compõem o processo de Mineração. Para alcançar o segundo objetivo, foi desenvolvida uma aplicação em Java que utiliza algoritmos de mineração como auxílio a um Sistema de Recuperação de Imagens por Conteúdo (SRIC), incorporando o conhecimento gerado no cálculo da similaridade entre imagens.

ABSTRACT

The growing facility to obtain, generate and store images leads to the need of new applications related to query, retrieval, analysis and discovery of knowledge from large databases. In order to extract useful knowledge, it is necessary to develop systems and algorithms that work with huge amounts of data and solutions that are able to classify, find patterns and relationships efficiently. In this context, a new technology arises, Image Mining, combining techniques from different domains. This work has two main tasks or objectives: (1) to present a survey on Image Mining and its relationships with other disciplines, and; (2) to model and develop an Image Mining Tool and Architecture to be applied on a range image database. To accomplish the first task, a review is presented for some Data Mining and Machine Vision concepts and algorithms, as well as the phases that compose an Image Mining process. In order to perform the second task, a Java application that uses mining algorithms is developed to help a Content-Based Image Retrieval System (CBIR), using knowledge to compute similarity between images.

1 INTRODUÇÃO

É grande a quantidade de informações disponíveis no momento, não apenas textos, mas também imagens, vídeo e áudio, isto é, dados multimídia. Essa informação está cada vez mais acessível, tendo a Internet como catalisadora, mas também é acompanhada de avanços científicos que naturalmente geram ainda mais dados, tais como pesquisas que envolvem Visão Computacional, Medicina ou Sensoriamento Remoto. Como consequência, diversas tecnologias têm evoluído para suprir a necessidade de extrair algo útil e resumido a partir dessas informações, ou seja, surge a necessidade de se descobrir conhecimento [Adriaans & Zantinge 96].

Cada técnica, metodologia ou disciplina criada é especializada na solução de um problema. Para organizar, recuperar e atualizar a informação são utilizados Bancos de Dados. Considerando apenas informações textuais, para extrair automaticamente informação das bases de dados com o mínimo de interferência do usuário, técnicas de Aprendizado de Máquina, Matemática e Estatística são unidas. Essas técnicas terão apoio de Sistemas Especialistas e estruturas de Inteligência Artificial para representar o conhecimento e utilizá-lo para extrair novos conhecimentos, dando origem à disciplina de Mineração de Dados.

Já do lado das informações multimídia, surgem bancos de dados que armazenam informações de diferentes tipos de estrutura, como som, imagem e dados espaciais. A imagem, por sua vez, precisa ser processada adequadamente antes de ser armazenada, passando assim por várias fases de pré-processamento, segmentação e representação. O resultado é uma enorme quantidade de imagens que precisam de alguma forma serem consultadas e recuperadas, fazendo uso assim de Sistemas de Recuperação de Imagens por Conteúdo (SRIC). Com funcionalidade próxima a um SRIC, encontram-se os Sistemas de Visão Computacional, cujo foco está na análise e interpretação de imagens com o objetivo, por exemplo, do reconhecimento de objetos e cenários.

Entretanto, apenas consultar os dados multimídia não traz novo conhecimento de maneira rápida, principalmente porque a quantidade a ser analisada é impraticável para o ser humano. No caso de dados mais complexos, tais como texto, sons, vídeo, imagens, documentos e modelos tridimensionais, surge a necessidade da Mineração de Dados Multimídia. Como interseção de várias terminologias, a Mineração de Dados Multimídia vem a ser uma área extremamente nova e desafiante, em especial a Mineração de Imagens.

Uma das motivações para a Mineração de Imagens está em introduzir novas abordagens às áreas e às disciplinas com as quais se relaciona, como é o caso das atuais implementações de SRIC e de Sistemas de Visão Computacional. Esses últimos ainda utilizam como parâmetros de pesquisa apenas aspectos de baixo nível de uma imagem (cor, textura e forma), o que é classificado como nível 1 (busca por características primitivas) por [Eakins 02]. Implementações novas ou melhorias desses mesmos sistemas vão além, também realizando a pesquisa por atributos derivados ou características lógicas (nível 2). Também em [Eakins 02], o autor aponta que a busca por atributos de baixo nível e de nível intermediário de uma imagem tem se mostrado deficiente, gerando desapontamento nos usuários.

Para que o processo de reconhecimento e interpretação de imagens seja eficiente, é necessário que as características extraídas e suas representações sejam afetadas o mínimo possível por variações como mudanças de tamanho, rotação e translação [Suk & Bhandarkar 92]. Uma das principais causas dessa deficiência é a chamada lacuna (*gap*) semântica [Gudivada & Raghavan 95], que é a falta de coincidência entre a informação extraída das imagens e a interpretação feita pelo usuário. Em [Smeulders et al. 00], é sugerido que, para diminuir essa lacuna, inclua-se o aprendizado de modelos semânticos, o que pode ser uma nova e interessante abordagem. Esse requisito vem preencher o nível 3 da classificação dos tipos de pesquisa de SRIC proposto por [Eakins 02], que seria a busca de conceitos abstratos, de alto nível ou de semântica.

Assim, torna-se clara a utilidade da Mineração de Dados como ferramenta integrante de um SRIC e de um Sistema de Visão Computacional, que além de tudo ainda provê desempenho e robustez na análise das informações. No caso de Visão Computacional, espera-se que o conhecimento obtido seja aplicável a diversas

classes ou tipos de objetos, não havendo ambigüidades ou redundâncias na representação dos mesmos.

1.1 OBJETIVOS

O primeiro objetivo desse trabalho é apresentar o tema Mineração de Imagens e demonstrar que não se trata apenas de uma nova área ou tecnologia, mas sim de um conjunto de ferramentas e algoritmos, com grande aplicabilidade na recuperação, análise e extração de conhecimento e semântica a partir de uma base de imagens. O outro objetivo é a concepção de um sistema de Mineração de Imagens genérico, que irá atuar como ferramenta de apoio a um SRIC, melhorando sua capacidade de consulta [Carston et. al 99] e procurando reduzir a distância entre características de baixo nível e de alto nível de uma imagem.

Mais especificamente, o aplicativo de Mineração de Imagem desenvolvido irá ser usado para a classificação de imagens de profundidade armazenadas em um banco de dados. A implementação, contudo, será feita de modo a deixar bem separada a funcionalidade do módulo de mineração, para que posteriormente esse mesmo aplicativo possa ser usado em outros sistemas, inclusive atuando como um componente do SRIC3D (Sistema de Recuperação de Imagens por Conteúdo de réplicas digitais em 3D de objetos físicos), projeto em desenvolvimento pelo grupo IMAGO¹.

1.2 ORGANIZAÇÃO

Na Seção 2, serão apresentados os conceitos básicos e o relacionamento da Mineração de Imagens com Banco de Dados, Processamento de Imagens e

¹ IMAGO - Grupo de Computação Gráfica e Processamento de Imagens da UFPR - <http://www.inf.ufpr.br/imagen>

Visão Computacional. Na Seção 3, serão detalhados o processo de Mineração de Imagens e suas etapas, além de aspectos de modelagem e indexação. Na Seção 4, serão vistas as principais abordagens e estruturas para extração de conhecimento, como classificação, agrupamento e descoberta de relações. Na seqüência, na Seção 5, serão descritos os principais algoritmos de Mineração de Dados e exemplos de aplicação em imagens. Na Seção 6, o assunto discutido será a classificação das arquiteturas de um Sistema de Mineração de Imagens, isto é, como modularizá-los ou integrá-los a outros sistemas. Para ilustrar os conceitos apresentados, na Seção 7, será discutido brevemente o projeto SRIC3D, que permitirá a pesquisa e mineração de uma base de imagens de profundidade [Hoover 98] e de modelos tridimensionais. Na seção 8 será detalhada a arquitetura do sistema de Mineração de Imagens desenvolvido, seus componentes, a base de dados minerada e os resultados experimentais. Com base nesses resultados, na arquitetura proposta e nos conceitos discutidos, na seção 9 serão apresentados trabalhos futuros e na seção 10, a conclusão.

2 CONCEITOS

Em [Zhang et al. 01], os autores definem que a Mineração de Imagens "cuida da extração de conhecimento implícito, relacionamento de imagens e outros padrões não explicitamente armazenados em bases de dados de imagens". Esse conceito simplifica o que a Mineração de Imagens tem a oferecer, negligenciando sua aplicabilidade em Visão Computacional e SRIC, bem como seu relacionamento com a Mineração de Dados. Como consequência, não fica evidente o porquê da criação de uma nova terminologia. Esta Seção tem como objetivo buscar uma definição mais ampla de Mineração de Imagens, deixando clara as diferenças e semelhanças com cada disciplina a qual ela se relaciona, tomando como partida a Figura 1, exibida na próxima página.

A Mineração de Imagens pode ser vista como um esforço multidisciplinar [Burl et al. 99] que une várias tecnologias. Nesse trabalho, o destaque será a integração entre Visão Computacional, SRIC, Mineração de Dados e Bancos de Dados. Porém, é preciso definir cada um dos termos e seus relacionamentos individualmente, para poder compor com precisão este novo conceito.

2.1 MINERAÇÃO DE DADOS

Mineração de Dados, também conhecida como KDD (*Knowledge Discovery in Databases*) ou *data mining* [Fayyad et al. 96b], é de fato a principal fase de um KDD, processo que envolve integração de dados, pré-processamento, mineração e pós-processamento [Adriaans & Zantinge 96]. Um KDD procura automatizar ao máximo a análise de dados feita pelo homem, isto é, o teste de hipóteses contra uma base de dados. O termo "ao máximo" indica que OLAP (*On-Line Analytical Processing*), por exemplo, é uma técnica de KDD que precisa de intervenção humana, pois é baseada na execução de consultas em grandes volumes de dados agregados e multidimensionais [Hasan & Hyland 01, Pedersen & Jensen 01].

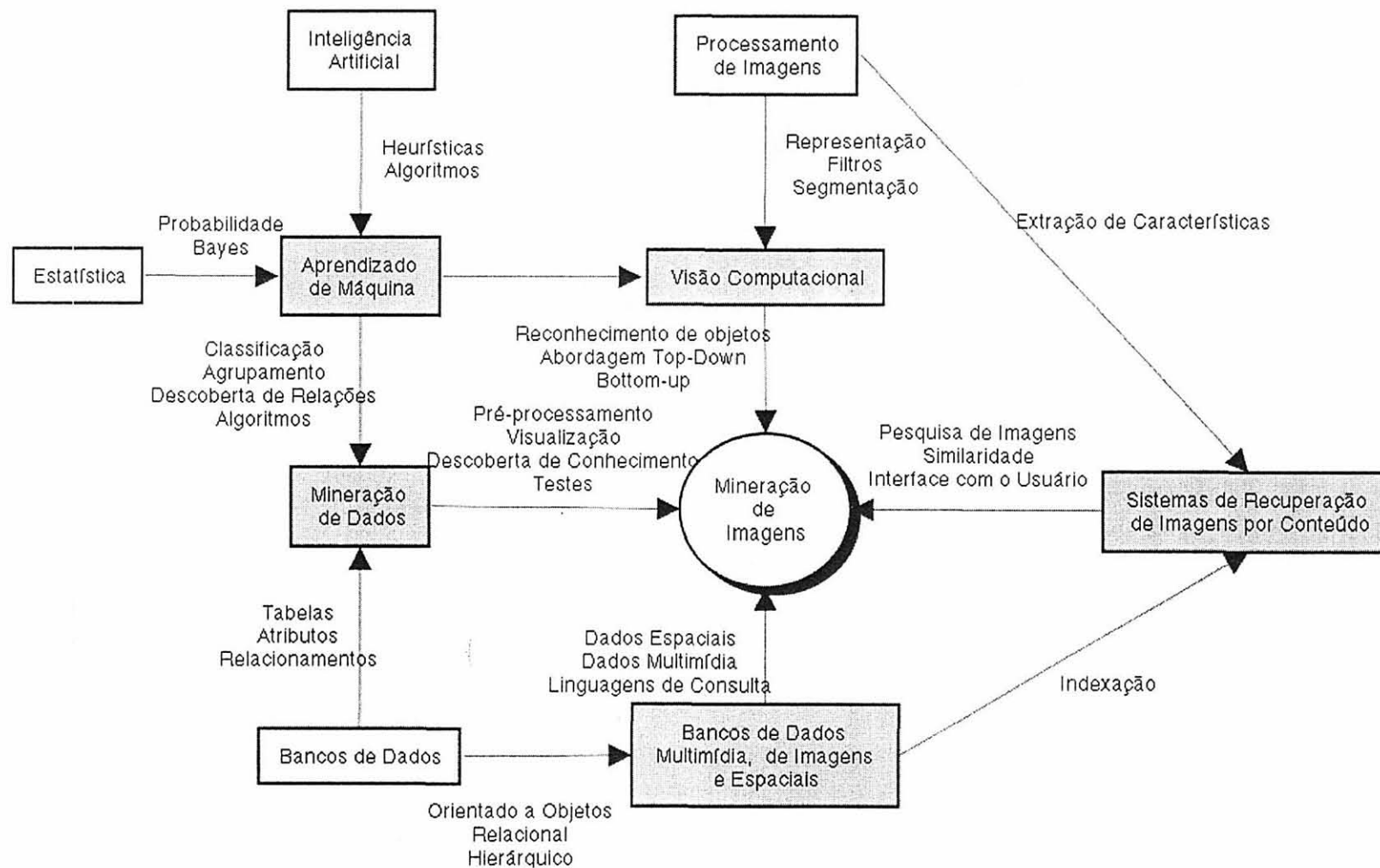


FIGURA 1 - INTEGRAÇÃO DA MINERAÇÃO DE IMAGENS COM OUTRAS TECNOLOGIAS

Conforme a Figura 1, a mineração de dados depende da contribuição de algoritmos e metodologias de Banco de Dados e Aprendizado de Máquina. Não é comum associar Banco de Dados com os algoritmos de baixo nível normalmente utilizados em processamento de imagens, mas é imprescindível falar de Banco de Dados quando o assunto é SRIC ou bases de imagens. Já aprendizado de máquina é de grande utilidade para guiar algoritmos de segmentação e interpretação dentro do contexto de Processamento de Imagens.

2.1.1 Banco de Dados.

Uma das diferenças entre Mineração de Dados e Aprendizado de Máquina é a preocupação que a mineração tem com a quantidade de dados a ser pesquisada. Por isso, a Mineração de Dados depende dos mecanismos de pesquisa, armazenamento e implementação usados em Sistemas Gerenciadores de Bancos de Dados (SGBD). Esses sistemas possuem ferramentas de *backup*, linguagens de consulta, controle transacional, controle de integridade, mecanismos de segurança e estruturas de indexação para permitir flexibilidade, robustez e performance no acesso a dados [Korth & Silberschatz 95]. Em especial, determinados aspectos de um banco de dados são bastante relevantes para técnicas de mineração de dados e de imagens: paradigmas de banco de dados, linguagens de consulta, dimensões e indexação. Mais detalhes sobre esse assunto serão apresentados na Subseção 3.3.

2.1.2 Aprendizado de Máquinas.

Definir Aprendizado de Máquinas envolve não apenas aspectos matemáticos, mas também filosóficos que fogem do escopo desse trabalho. Em [Mitchel 97], define-se que uma máquina aprende a partir de uma experiência E com respeito a alguma classe de tarefas T e medida de performance P , se a performance na tarefa T , medida por P , melhora com a experiência E . Em [Witten & Frank 00], os autores mostram uma visão mais prática do assunto, considerando que o objetivo do

aprendizado de máquinas está em procurar e descrever padrões estruturais de dados, explicar os dados e fazer previsões a partir dos mesmos.

Como pode ser visualizado na Figura 1, para realizar essas tarefas, o Aprendizado de Máquinas irá fazer uso de inúmeras outras disciplinas, principalmente Inteligência Artificial e Estatística [Mitchel 97]: da primeira, usará as heurísticas já implementadas, as estruturas de representação de conceitos e de conhecimento e a abordagem do Aprendizado de Máquina como um problema de busca; da segunda, tomará emprestado os testes estatísticos, a caracterização de erros, o teste de hipóteses e o cálculo de probabilidade. Expandindo esse domínio de relacionamentos, é possível ver que o Aprendizado de Máquina também faz uso de Teoria da Computação (complexidade), Filosofia (hipóteses), Teoria da Informação (entropia), Teoria de Controle (processos), Psicologia e Neurobiologia (redes neurais) [Mitchel 97].

2.2 VISÃO COMPUTACIONAL

Quando se discute Inteligência Artificial com Processamento de Imagens, o conceito a ser utilizado é o de Visão Computacional. Este conceito envolve várias atividades, desde a captura e processamento das imagens até o reconhecimento de objetos de um cenário, como também a interpretação de uma imagem em conjunto com os aspectos psicológicos e fisiológicos da visão. Mesmo com essa distância entre o conhecimento para modelar e descrever o mundo e a representação das imagens, o desafio da compreensão das imagens permanece e é bastante explorado por Sistemas de Visão Computacional.

Para [Jain et al. 95], o objetivo da Visão Computacional é criar um modelo do mundo real por meio da extração de informações úteis a partir das projeções bidimensionais de uma cena através de um processo composto das seguintes fases:

- a) melhoria da imagem através de técnicas de processamento de imagem;
- b) extração de características por técnicas de segmentação e representação;
- c) relacionamento das características com a geometria dos objetos, unindo primitivas de computação gráfica e técnicas de reconhecimento de padrões, e;
- d) interpretação das informações geométricas obtidas - regiões, padrões, objetos - usando heurísticas de inteligência artificial com base em modelos psicofísicos da visão humana.

Um ponto chave da Visão Computacional é o uso de conhecimento prévio das imagens, em especial para reconhecimento de objetos em imagens reais. Em [Jain et al. 95], é descrita uma arquitetura de um Sistema de Visão Computacional para reconhecimento de objetos que é formado por: base de modelos, detetor de características, gerador e verificador de hipóteses. A base de modelos armazena descritores de objetos, denominados vetores de características. As características detectadas são usadas pelo gerador de hipóteses para atribuir um grau de semelhança entre os objetos e ainda são revisadas na fase de verificação. Como será visto, a Mineração de Imagens segue estrutura semelhante a proposta em [Jain et al. 95], porém enfatizando a atualização e evolução do modelo como auxílio ao processo de Visão Computacional. A proposta é que a mineração permita que o modelo seja revisto, conforme a necessidade, a cada novo exemplo, como resultado da fase de verificação de hipóteses.

2.3 SISTEMAS DE RECUPERAÇÃO DE IMAGENS POR CONTEÚDO

SRIC são sistemas responsáveis pela pesquisa, consulta e recuperação de dados visuais, normalmente utilizando atributos de baixo nível de uma imagem, tais como cor, textura, forma, e fazendo uso de informações semânticas (palavras-chave e anotações). Trata-se de uma ferramenta usada para pesquisa de um grande

número de imagens (e vídeo [Araújo & Guimarães 00]) em bancos de dados apropriados para esse tipo de consulta. Em [Pereira 01], é apresentado um levantamento detalhado sobre esses sistemas, bem como uma proposta de classificação.

Em conjunto, SRIC e Bancos de Dados Multimídia (vide Figura 1) trabalham no acesso ao conteúdo de uma base de imagens e precisam lidar com alguns aspectos, tais como as diferenças semânticas que envolvem a avaliação de uma imagem, a extração de suas características, a indexação das propriedades obtidas e o uso dessas informações para consulta, interpretação e reconhecimento de objetos. Além do processamento de imagens e armazenamento, os SRIC devem conter mecanismos de busca de imagens, seja através da navegação (*browsing*) ou através de consultas (*queries*), de maneira semelhante ao que ocorre em bancos de dados textuais [Rui et al. 99]. A navegação simples sobre as imagens não traz vantagem se a base de imagens for grande, pois se trata apenas da exibição de cada imagem cadastrada no sistema. Mesmo assim, a navegação do resultado de uma consulta não pode ser ignorada, pois configura-se como uma ferramenta de interatividade do usuário.

Um dos grandes desafios dos SRIC denomina-se diferença semântica (*semantic gap*) [Gudivada & Raghavan 95], que é a falta de coincidência entre o que os sistemas provêm e o que o usuário deseja. De posse das características extraídas de uma imagem, bem como informações obtidas de outros processos de segmentação e também de processos manuais, a distância semântica se mantém e, pior, em conjunto com a distância sensorial, que é a falta de uniformidade nas condições de aquisição das imagens, como luminosidade, câmera e qualidade [Smeulders et al. 00]. É na fase de interpretação que "*gaps*" serão tratados, seja através de funções que calculam a similaridade dos conteúdos das imagens ou incorporando mecanismos de aprendizado. A interpretação através de conceitos de similaridade trabalha com a segmentação de imagens para obter características globais e locais, para em seguida, desenvolver heurísticas que estabelecem uma função que permite obter um grau de semelhança entre as imagens.

Um outro tipo de similaridade que não trabalha somente com atributos de baixo nível e nível intermediário é a semântica. Em [Smeulders et al. 00], são citados

trabalhos que, a partir de algoritmos de *clustering* e de dicionários de textura, geram uma base de conhecimento que é usada pelo usuário para pesquisa de imagens. Essas aplicações atuam em conjunto com processos interativos e iterativos de manipulação de pesquisas e com técnicas de realimentação de relevância (*relevance feedback*), que permitem que o usuário informe qual a importância do resultado de cada imagem retornada [Eakins 02] e assim refine a base de conhecimento do sistema. Com isso, a informação de contexto de uma imagem e a sua semântica podem ser aprendidas sem necessidade que o usuário classifique todas as imagens disponíveis, uma vez que o sistema será capaz de adaptar-se com a experiência, como em sistemas tutores [Direne 97].

2.4 DEFINIÇÃO DE MINERAÇÃO DE IMAGENS

Para [Zhang et al. 01], o foco da Mineração de Imagens é "determinar como a representação de baixo nível (*pixels*) de uma imagem pode ser eficientemente e efetivamente processada para identificar objetos e seus relacionamentos". Enquanto SRIC se preocupam em consultar e recuperar grandes quantidades de imagens e a Visão Computacional e o Processamento de Imagens visam extrair informação de uma imagem, a mineração de imagens trabalha com a recuperação, consulta e análise inteligente de imagens. Em [Zhang et al. 01], considera-se que as técnicas e algoritmos tradicionais de Mineração de Dados precisam ser adaptados para a realidade de imagens devido a vários fatores:

- a) valores relativos e absolutos: um campo numérico numa tabela de banco de dados possui um valor estante, ao passo que em uma imagem depende dos valores que estão próximos, tendo um significado semântico diferente;

- b) informação espacial: bancos de dados relacionais precisam ser adaptados para guardar esse tipo de informação;
- c) múltiplas interpretações: um texto ou uma *string* possuem interpretações conhecidas, mas uma imagem depende totalmente de sua contextualização;
- d) representação de padrões: a representação deve saber tratar justamente do contexto e das informações espaciais;
- e) seleção de características: é preciso saber escolher quais são as características mais significativas, e;
- f) visualização de padrões: é preciso criar novas formas de visualização dos padrões encontrados.

Os autores [Ordonez & Omiecinski 99] afirmam que "a tecnologia de descoberta de conhecimento atual está longe de ser capaz de extrair todo o conhecimento contido em dados multimídia". Em [Djeraba 00], o autor estabelece uma questão que vem resumir os desafios da Mineração de Imagens: "como um sistema pode extrair não apenas características visuais mas também relacionamentos escondidos entre características com o objetivo de permitir uma busca semântica?". A solução está justamente no que foi abordado no tópico anterior: aprendizado com experiência, extração automática de conhecimento e realimentação de relevância.

3 PROCESSO DE MINERAÇÃO DE IMAGENS

Como já citado, a Mineração de Dados e de Imagens fazem parte de um processo maior chamado KDD, que vai da obtenção dos dados a serem analisados, descoberta de padrões, sua aplicação e validação. Segundo [Fayyad et al. 96b], esse processo é dividido nas seguintes etapas:

- a) identificação do objetivo;
- b) criação do conjunto de dados alvo, isto é, atributos e exemplos de dados;
- c) limpeza dos dados, retirada de ruídos, análise dos atributos, análise de dados incompletos e pré-processamento;
- d) redução dos dados e projeção, ou seja, reduzir o número de variáveis envolvidas;
- e) escolha da técnica para extração do conhecimento;
- f) escolha do algoritmo;
- g) mineração de padrões;
- h) interpretação, e;
- i) consolidação do conhecimento, documentação e avaliação.

A Figura 2 propõe que, no caso de imagens, também é possível adicionar mais atividades ao processo de KDD, tais como etapas importantes das áreas de Visão Computacional, SRIC e Processamento de Imagens, como será visto a seguir.

3.1 OBTENÇÃO

A obtenção dos dados é a fase em que eles são escolhidos com base no objetivo a ser atingido. No caso de imagem, essa fase é mais evidente, pois se inicia

quando a imagem é adquirida. Outra possibilidade, é aproveitar bases de imagens já existentes, como as obtidas facilmente a partir da Internet. Também é válido combinar dados de outras naturezas e de outras origens, como dados médicos ou geográficos.

3.2 PRÉ-PROCESSAMENTO

Os dados a serem minerados vêm de diversas bases, por vezes incompletas, ruidosas e inconsistentes. Conforme a Figura 2, antes de efetuar a integração, os dados devem passar por procedimentos de limpeza, seleção e transformação [Adriaans & Zantinge 96]. A limpeza é a eliminação ou filtragem de dados inconsistentes. A seleção consiste na eliminação de atributos redundantes, registros duplicados e redução da dimensão dos atributos. Esse processamento pode ocorrer com os dados armazenados no banco de dados e também em cada imagem antes da integração, através de técnicas de redução de ruído, suavização e aplicação de filtros [Gonzales & Woods 92].

Ao mesmo tempo em que os dados são filtrados e selecionados, também é necessário adaptá-los às necessidades dos algoritmos de mineração de dados. Por exemplo, não é comum trabalhar diretamente com mineração de *pixels*, ou seja, minerar diretamente a representação matricial de uma imagem. Dessa maneira, como parte do pré-processamento, é interessante converter as imagens e as suas características para um esquema de representação adequado, isto é, tabelas, registros, atributos e estruturas de dados. Essa fase será vista com mais detalhes no tópico 3.5, sobre a modelagem do banco para mineração de imagens.

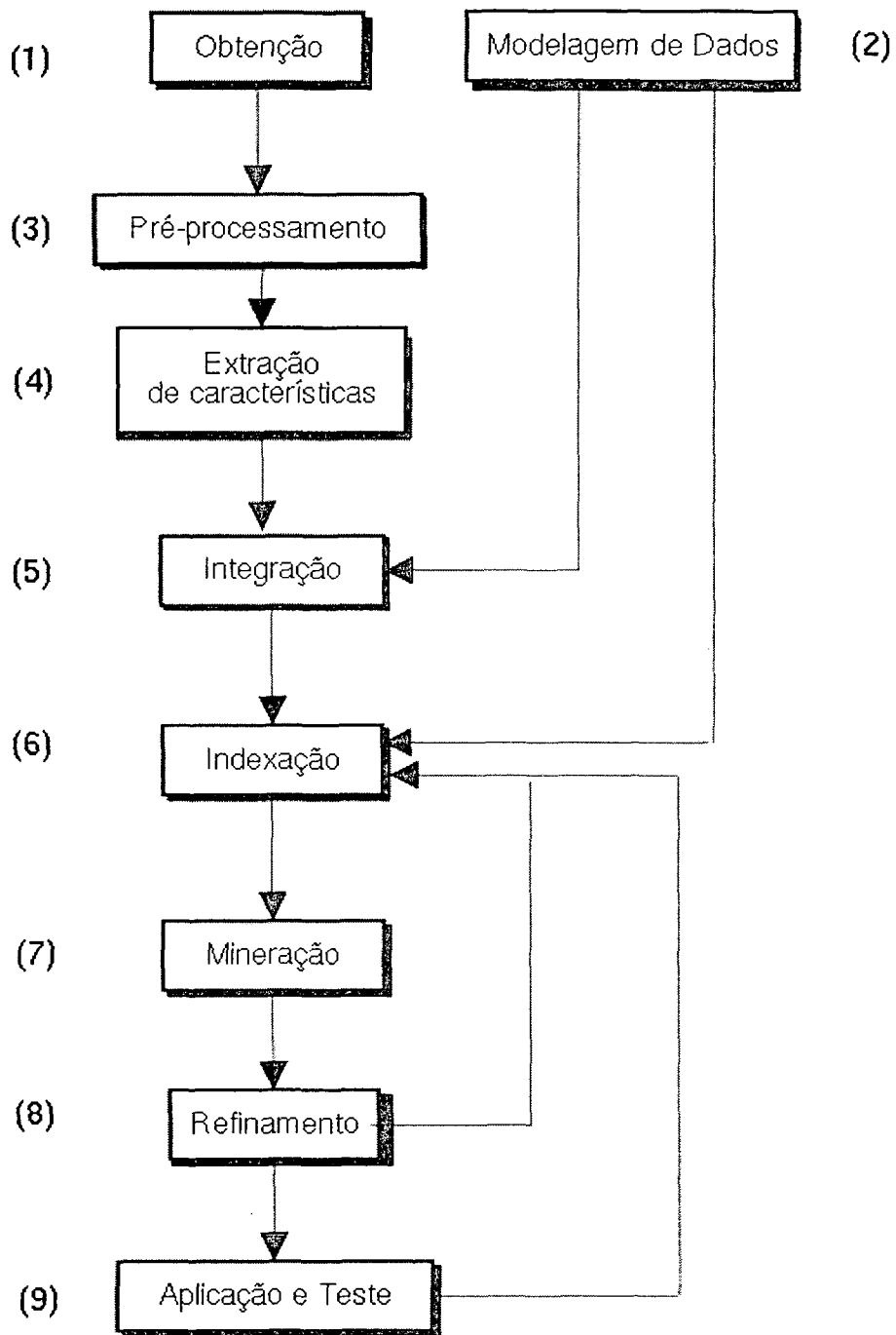


FIGURA 2 - PROCESSO DE MINERAÇÃO DE IMAGENS

(1) Aquisição de diferentes fontes; (2) Representação das características em Bancos de Dados; (3) Melhoria das imagens; (4) Características; (5) Armazenamento; (6) Estrutura para consultas; (7) Aplicação dos algoritmos de aprendizado de máquina; (8) Visualização e representação do conhecimento; (9) Aplicação e verificação do conhecimento

3.3 EXTRAÇÃO DE CARACTERÍSTICAS

Essa etapa é exclusiva da mineração de imagens. Como já apresentado, não há interesse em trabalhar diretamente com *pixels* e sim com uma representação intermediária. Essa abordagem é utilizada em SRIC, que trabalham com características de baixo nível como cor, forma e textura, representadas através de vetores de características. Em [Vailaya 00], afirma-se que os métodos para extração de características se dividem em dois grupos básicos: os que preservam a informação espacial (como CSS ou *Curvature Scale Space* [Mokhtarian et al. 96]) e aqueles que não a preservam (como histogramas). Três abordagens são definidas em [Khoshafian & Barei 96] para extração de características:

- a) extração manual, através da entrada de dados feita pelo usuário;
- b) extração automática, através de métodos de reconhecimento de padrões, segmentação e outros algoritmos, e;
- c) extração híbrida, em que o sistema extrai as informações e o usuário as valida.

Para análise de cor, os principais métodos são o histograma, os momentos e as matrizes de co-ocorrência [Smeulders et al. 00]. Ainda segundo [Vailaya 00], a extração de cor descreve propriedades globais de uma imagem e podem ser facilmente extraídas, são invariantes quanto a alterações de tamanho da imagem, mas não incluem informações espaciais. A característica forma permite obter informações locais e geométricas, intolerantes a variações de escala, orientação e translação, tendo como principais exemplos de técnicas a aproximação poligonal, os momentos invariantes e os descritores de *Fourier* [Smeulders et al. 00]. Textura, conforme definida por Haralick [Haralick et al. 95], é um conjunto de propriedades de uma imagem, tais como uniformidade, densidade, aspereza, regularidade e intensidade, obtidas por meio da análise da matriz de co-ocorrência, M-SAR e *wavelets* [Smeulders et al. 00]. Na prática, os SRIC combinam as várias classificações de características para melhorar a eficiência das consultas.

3.4 INTEGRAÇÃO

De posse das imagens pré-processadas e das características geradas, é preciso consolidar as informações obtidas nas tabelas e atributos definidos pela fase de modelagem. Essa fase de consolidação é fortemente ligada à atividade de pré-processamento, envolvendo a redução de dimensão ou o uso de métodos de transformação para a redução do número de variáveis que representam os dados [Fayyad et al. 96b]. Como redução de dimensão, entende-se a escolha dos atributos que melhor representam um registro, imagem ou objeto, permitindo diminuir a quantidade de dados que passará pelas fases de indexação e mineração. No domínio de imagens, uma solução prática para redução de atributos é a transformada Karhunen-Loève, que é uma técnica matemática usada para análise de variáveis randômicas correlacionadas que pode ser usada para descartar características redundantes [Nadler & Smith 93].

3.5 MODELAGEM DO BANCO DE DADOS

A fase de modelagem é uma atividade de análise dos dados e do objetivo do sistema que deve anteceder ou ocorrer em paralelo às fases de integração e indexação. Para que ocorra a mineração de dados, é necessário que exista um conjunto de dados em que cada registro se constitui em uma instância, e cada instância é composta de atributos, campos ou características. Assim, é preciso adaptar os dados das etapas anteriores para que possam ser minerados. Uma abordagem de mineração consiste em reunir dados distribuídos em várias tabelas para que sejam convertidos em uma única tabela, composta de registros ou exemplos independentes, individuais, sem relações com os demais, conhecidos como instâncias [Witten & Frank 00]. Em contrapartida, no trabalho de [Siromoney et al. 00], a ferramenta de mineração utilizada permite que vários tipos de exemplos sejam informados, sem a necessidade de convertê-los em instâncias independentes.

Em [Witten & Frank 00], os autores classificam os atributos de uma instância em nominais ou contínuos. Os nominais possuem um número finito de valores possíveis, cada um associado a um símbolo. Exemplos de atributos nominais são facilmente encontrados em bancos de dados textuais, tal como o campo sexo (masculino, feminino). Os atributos nominais também podem ser chamados de enumerados ou discretos, pressupondo uma ordenação e a associação com uma grandeza numérica. Atributos numéricos, contínuos ou ordinais são bastante aplicáveis no caso de imagens, pois as características são em sua maioria valores numéricos. Técnicas de mineração que envolvem estatística, são capazes de trabalhar diretamente com esses tipos de valores. Um atributo numérico está associado a uma escala ou intervalo. Um exemplo é estipular intervalos de idade para definir se uma pessoa é criança, jovem, adulta ou idosa.

No que se refere a paradigmas, há vários tipos de bancos de dados. Também é possível classificar um banco segundo o tipo de dado usado, como dados estruturados e dados semi-estruturados. Nos próximos itens, serão vistas algumas abordagens de como organizar os dados conforme a sua natureza para aplicar as técnicas de mineração de dados, tais como: estrutura relacional, estrutura orientada a objeto, bases multidimensionais, dados semi-estruturados, bases multimídia e bancos de dados espaciais. A Figura 3 classifica e exemplifica os bancos de dados segundo a sua natureza, estrutura e aplicação.

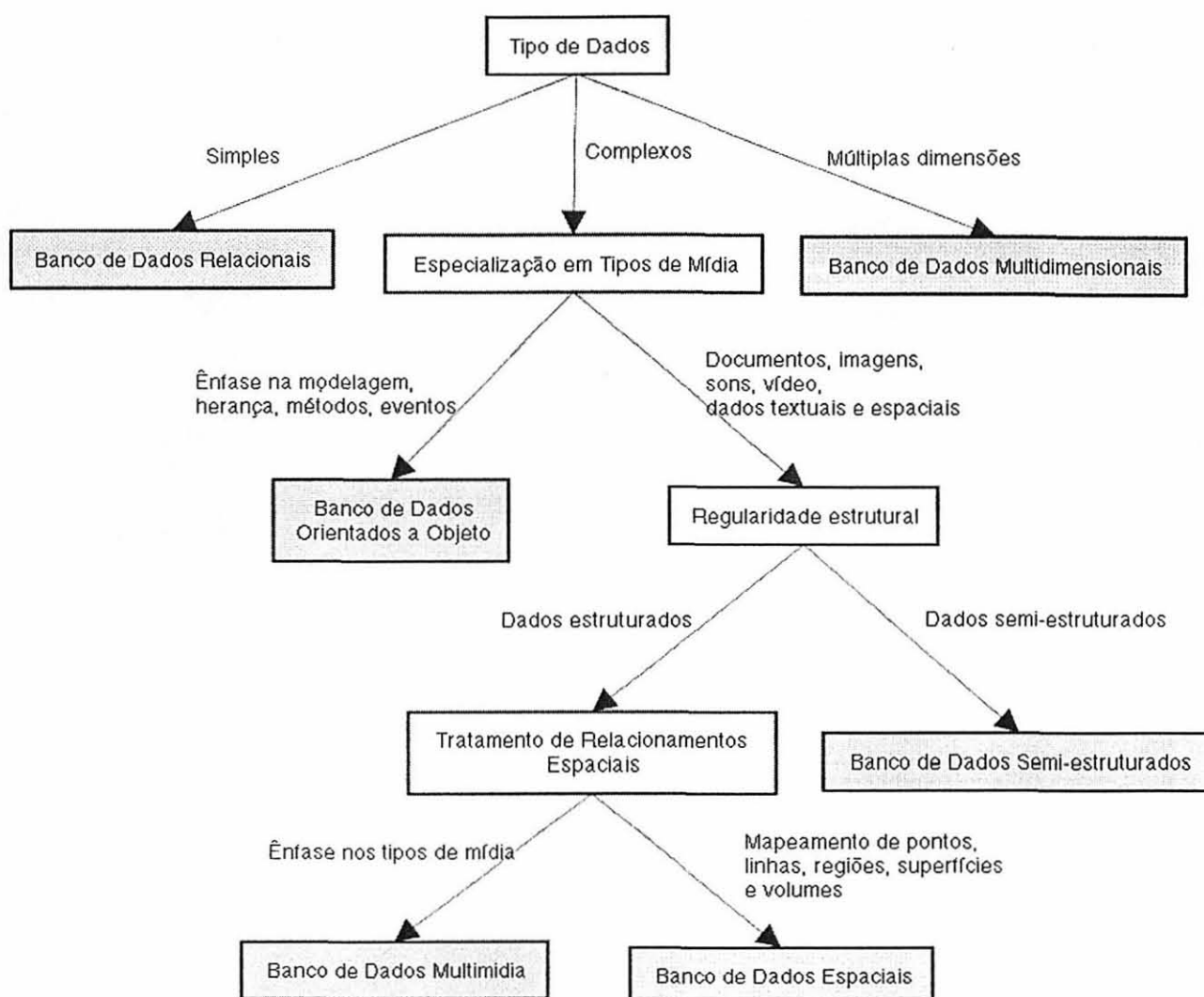


FIGURA 3 - PRINCIPAIS ESTRUTURAS DE BANCO DE DADOS [Vieira 01]

3.5.1 Bancos de dados relacionais e orientados a objeto.

Um banco de dados relacional usa lógica e teoria de conjuntos para organizar dados em tabelas, que contêm um conjunto de registros, sendo que cada registro possui campos [Korth & Silberschatz 95]. Trata-se da implementação mais popular atualmente, presente na maioria dos bancos de dados, tais como Oracle,

Sybase, Microsoft SQL Server ou PostgreSQL². Do outro lado, encontram-se os bancos orientados a objetos que seguem o paradigma de orientação a objetos, tratando os dados como objetos, compostos por atributos, propriedades e métodos [Kim 95].

Um banco de dados relacional é mais recomendado para bases de dados comuns e estruturadas, envolvendo campos no formato texto ou numérico, como cadastros de pessoas ou atributos simples de uma imagem. Um banco de dados orientado a objeto permite uma flexibilidade maior na modelagem de dados, incluindo o uso de estruturas de dados mais híbridas, englobando objetos complexos, dados comportamentais e metaconhecimento [Korth & Silberschatz 95].

3.5.2 Bancos de dados multidimensionais.

A mineração de um cubo de dados é um exemplo clássico de mineração de bases multidimensionais, como visto no sistema MultimediaMiner [Han et al. 97]. Um cubo de dados ou hipercubo pode ser visto como um vetor multidimensional em que cada dimensão representa um atributo e cada célula armazena um atributo agregado, como soma, média, total [Zaiane et al. 98].

3.5.3 Bancos de dados semi-estruturados.

Em [Abiteboul 01], o autor define dados semi-estruturados como aqueles que apresentam um grau de regularidade menor do que um banco de dados relacional ou orientado a objetos. Um dado pode ser visto como não estruturado no início de um processamento mas ao final poderá apresentar uma certa estrutura. Um exemplo é o banco de dados de XML³ (eXtensible Markup Language), que de uma maneira geral pode ser visualizado como uma árvore composta de atributos e por nós que podem ter um número desconhecido de folhas ou outras ramificações. Esse tipo de banco de dados é citado nesse trabalho devido a sua aplicabilidade em

² <http://www.postgreSQL.org>

³ <http://www.w3.org/xml>

certos tipos de mineração de dados, em especial de dados multimídia e de conteúdo da Web [Khoshafian & Barei 96].

3.5.4 Bancos de dados multimídia e imagens

Como já visto, as bases de imagens fazem parte de uma disciplina maior, que é a de Banco de Dados Multimídia. Mais precisamente: “bancos com gerenciadores de bases de dados de alta capacidade e performance que suportam vários tipos de multimídia, como também tipos alfanuméricos e manipulam grandes volumes de informação” [Khoshafian & Barei 96]. No caso de imagens, já existem implementações de linguagem de consulta específicas para esse domínio, IMQL (*Image Query Language*) [Khoshafian & Barei 96]. Extensões próprias para imagens podem ser vistas em alguns bancos de dados, como o PostgreSQL, que contém funções específicas para realizar cálculos em imagens e dados espaciais (PostGIS⁴).

3.5.5 Bancos de dados espaciais.

Como será visto, a aplicação de mineração de dados será feita sobre uma base que combina características de baixo nível e nível intermediário de uma imagem, além de dados espaciais obtidos das regiões segmentadas, tais como vértices de polígonos que representam o contorno das regiões e o relacionamento entre as mesmas. Essa última abordagem coincide com uma das aplicações possíveis dos bancos de dados espaciais. Conforme [Samet 95], dados espaciais são objetos compostos por pontos, linhas, regiões, retângulos, superfícies, volumes e até dados de dimensões maiores.

⁴ <http://postGIS.refrains.net>

3.6 INDEXAÇÃO

Tanto em bancos de dados tradicionais quanto multimídia e espaciais, é preciso que o acesso ou consultas aos registros armazenados seja feito de forma eficiente. Vários métodos de indexação são descritos por [Del Bimbo 99], envolvendo imagens, vistos a seguir, mas também ressalta a importância de combinar similaridade perceptual e similaridade visual. Assim, a pesquisa no banco de imagens seria uma composição de índices para dados textuais (árvores, tabelas *hash* [Sedgewick 98]) e índices para dados visuais das imagens. O fator de similaridade na pesquisa de uma imagem é uma composição dos pesos desses dois tipos de dados.

Em [Khoshafian & Barei 96], os autores separam a indexação em 2 tipos: métodos e estratégias. O primeiro “ênfatiza o *quê* indexar” e o segundo, “*como* indexar”. A indexação pode usar “*strings*” para representar uma estrutura espacial ou palavras-chave e anotações. Quanto às estratégias de indexação, a implementação pode ser feita usando estruturas multidimensionais que envolvem índices simples ou com vários atributos. As chaves podem ser associativa ou baseadas em *hashing*: a primeira é útil para pesquisas que envolvem limites ou graus de imprecisão; a outra é aplicada para consultas exatas. Em [Smeulders et al. 00], as principais estruturas de indexação multidimensional são divididas em três tipos: particionamento espacial, particionamento de dados e técnicas baseadas em distância. No primeiro tipo, as características são organizadas em estruturas de dados de árvore, como *k-d tree* ou *k-d B tree*. No segundo tipo, a cada ponto no espaço de características está associada uma região que representa a vizinhança de pontos, como nas estruturas *R-tree*, *R*-tree*, *R+-tree*, *VAM split R-tree* e *SR-tree* [Gaede & Gunther 98]. O terceiro tipo, composto por estruturas tais como *VP-tree* e *MVP-tree* [Bozkaya & Ozsoyoglu 97], divide o espaço de características em grupos de anéis concêntricos especificados por distâncias relativas a um ponto representativo (*vantage point*), sendo úteis para pesquisas baseadas em similaridade.

Existem outras classificações e cada uma tem sua peculiaridade quanto à divisão do espaço de busca e critérios de organização. A escolha de uma ou outra estrutura deve ser adequada ao tipo de dado que se deseja armazenar e como se

deseja recuperá-lo. Esse trabalho não visa escolher qual o melhor tipo de indexação tampouco avaliar as estruturas já existentes, contudo o impacto da escolha no processo de mineração de imagens precisa ser levado em consideração na avaliação da performance e precisão das consultas.

3.7 MINERAÇÃO

A mineração é a principal fase de um sistema de KDD. A metodologia, as técnicas e os algoritmos de mineração de dados serão vistos com mais detalhes nas Seções 4, 5 e 6.

3.8 REFINAMENTO

A fase de refinamento é necessária para melhorar a legibilidade do conhecimento gerado e compreendê-lo para aplicá-lo na prática. A mineração de dados pode representar o conhecimento extraído de diversas formas, como regras, árvores de decisão, agrupamentos, tabelas ou redes neurais [Azevedo et al. 00]. Se a quantidade de regras ou o tipo de estrutura usada não permitir uma leitura trivial, é necessário fazer uso de técnicas de visualização ou a conversão dessas estruturas complexas, como redes neurais, em regras facilmente interpretáveis [Hayashi et al. 00].

3.9 APLICAÇÃO E TESTE

O próximo passo é aplicar o conhecimento gerado em uma base de dados e efetuar a avaliação dos resultados, para em seguida atualizar ou corrigir o conhecimento gerado, refazer o processo, e executar novamente as consultas,

enfim, realimentar o sistema. Os SRIC possuem uma etapa semelhante, em que interatividade do usuário não se resume em apenas consultar o resultado final de uma pesquisa. Por meio de interfaces de usuário mais avançadas, o próprio conjunto resultado de uma consulta pode permitir interações mais avançadas. O usuário pode refazer a sua pesquisa tendo como ponto de partida o resultado da última consulta, bem como atualizar a base de conhecimento do sistema. Assim, a intervenção humana é necessária para refinamento de um KDD e também de SRIC.

A execução das etapas de um processo de KDD são necessárias não apenas para sugerir uma ordem de execução como também propõem uma maneira de modularizar o sistema. Se for feita uma análise sob o ponto de vista da etapa de mineração, as etapas anteriores são responsáveis pela obtenção e consolidação dos dados e as etapas posteriores são necessárias para conferência e evolução do sistema. A mineração propriamente dita consiste em estratégias para descrição ou classificação dos dados, como será visto adiante.

4 ESTRATÉGIAS DE MINERAÇÃO DE DADOS E IMAGENS

O objetivo desta Seção é apresentar as principais tarefas que o processo de mineração de dados e imagens pode desempenhar, que vem a ser o tipo de conhecimento que se deseja extrair de um banco de dados. Nesse item não são exemplificados algoritmos, mas tarefas que guiam a escolha do algoritmo utilizado e o tipo de pesquisa que o usuário deseja obter.

4.1 CLASSIFICAÇÃO

Das atividades de mineração de dados, é possível verificar que um de seus principais objetivos é a inferência de regras que permitam classificar dados, ou seja, a partir de um conjunto de treinamento são descobertas regras que podem ser aplicadas a novos conjuntos. Isso é semelhante à abordagem *bottom-up* de reconhecimento e interpretação de imagens, onde o modelo é construído a partir dos dados de baixo nível, conforme Seção 6.

Classificação consiste em associar um conjunto de atributos (tupla) a um atributo objetivo. De maneira mais formal, a tarefa de classificação é “descobrir algum tipo de relacionamento entre valores de predição e o valor objetivo, tal que o conhecimento descoberto possa ser usado para prever a classe de uma nova tupla”, [Freitas & Lavington 98]. Um exemplo simples de classificação consiste em, a partir de uma base de dados de avaliações ambulatoriais (exames e procedimentos), gerar um conjunto de regras que irão auxiliar no diagnóstico de alguma doença.

4.2 DESCOBERTA DE REGRAS DE ASSOCIAÇÃO

A descoberta de regras de associação não envolve apenas um atributo objetivo, como visto no tópico anterior. Uma regra de associação é uma expressão $X \Rightarrow Y$ aplicada sobre um conjunto de transações D , implicando que a cada transação T que contenha X , é provável que também contenha Y , respeitando critérios de confiabilidade e suporte [Hipp et al. 00, Kawahara & Kawano 01]. A confiabilidade é definida como a porcentagem de transações contendo $X \Rightarrow Y$ sobre o total de transações que contém X . O conceito de suporte é definido como a proporção entre o total de instâncias em que a regra de associação é válida sobre o total de transações D . Uma aplicação que envolve descoberta de regras de associação pode ser vista em [Wang & Chung 01], que descreve um sistema de recuperação de imagens que também possui um módulo de mineração de imagens de câncer de pele em conjunto com diagnósticos.

4.3 AGRUPAMENTO

Nessa atividade, o sistema de KDD deve induzir, de maneira não supervisionada, um esquema de classificação que particiona o conjunto de treinamento em classes [Celinski & Bellon 98, Celinski 98]. A partir das classes, será possível aplicar algoritmos de classificação ou sumarização. Essa idéia está bastante associada à predição de atributos ou invocação de modelos (vide Seção 2.2). Se por meio da classificação for descoberto que um padrão encontrado pertence a uma certa classe, os demais atributos desse padrão podem ser deduzidos. Mais detalhes de como funcionam os algoritmos de agrupamento podem ser vistos na Seção 5.

4.4 OUTRAS TAREFAS

Existem outras tarefas que também compõem a estratégia de mineração de dados:

- a) detecção de desvios: é uma tarefa que visa descobrir desvios entre o conteúdo atual e valores previamente medidos [Fayyad et al. 96b];
- b) modelagem de dependências: uma variação da classificação, em que normalmente o “antecedente e o conseqüente de uma regra podem ter um número variado de atributos” [Freitas & Lavington 98];
- c) regressão: uma função que mapeia um item de dados para uma variável numérica (contínua) [Fayyad et al. 96b];
- d) sumarização: tem como objetivo “produzir uma descrição característica de cada classe de tuplas no conjunto de dados alvo” [Freitas & Lavington 98], de maneira semelhante a um agrupamento, e;
- e) estimativa e previsão: consistem na determinação do valor mais provável ou o comportamento de algum índice diante de dados do passado ou de outros índices semelhantes sobre os quais se tem conhecimento [Carvalho 01].

É na escolha da estratégia que o usuário define como o sistema, de maneira supervisionada, aprenderá a classificar as instâncias de um banco de dados ou, de maneira não supervisionada, fará a geração de regras ou agrupamentos que descrevem os dados. Na estratégia de mineração é essencial o processo de descoberta do conhecimento, pois está intimamente ligada aos padrões que se deseja encontrar e ao algoritmo a ser utilizado. Contudo, não há uma regra que indique qual o melhor algoritmo, e um sistema deve ser flexível para prover a possibilidade de trocar ou combinar algoritmos de mineração.

5 ALGORITMOS E APLICAÇÕES

A seguir, serão descritos os princípios dos algoritmos mais comuns e algumas soluções mais recentes, oriundas de aplicações práticas de mineração de dados e imagens. Não se trata de um estudo completo, mas uma referência para a definição da arquitetura de um sistema de mineração de imagens.

5.1 ÁRVORES DE DECISÃO E INDUÇÃO DE REGRAS

Algoritmos de aprendizado de máquina são bastante utilizados em tarefas de classificação ou de descoberta de regras de associação. Uma árvore de decisão consiste de nós que são os testes dos atributos e cada ramificação determina os possíveis valores que o atributo pode assumir. Um teste pode dar origem a mais subdivisões ou, chegando a uma folha, classificar a informação sendo analisada. Para [Lavraç 99], o processo de construção de uma árvore de decisão é um processo heurístico que procura minimizar o número de testes necessários para classificação pela escolha do atributo mais significativo. Exemplos de algoritmos de indução de regras e de árvore de decisão são o ID3, ASSISTANT e C4.5 [Mitchel 97].

No SKYCAT, projeto de um sistema que provê um ambiente integrado para construção, classificação, gerência e análise de catálogos de imagens astronômicas, [Fayyad et al. 96a] utiliza variações dos algoritmos ID3 e C4.5: GUID3 e O-BTree. O primeiro algoritmo, GUID3, foi utilizado para diminuir o particionamento dos dados, criando subdivisões que representam mais de um valor. O outro algoritmo, O-Btree, utiliza outras medidas para seleção dos dados, ao invés de usar apenas o grau de entropia.

5.2 ALGORITMOS DE CLUSTERIZAÇÃO

Para análise de agrupamentos, podem ser usados algoritmos específicos, que podem ser classificados em [Carvalho 01, Celinski 98]: hierárquicos, aglomerativos, divisivos, não-hierárquicos e de partição. Esses algoritmos têm como objetivo "classificar, com respeito a algum critério pré-determinado, uma amostra de entidades em grupos mutuamente exclusivos baseado nas suas similaridades". O que define algoritmos hierárquicos ou divisivos é que "a união de dois grupos numa certa etapa produz um dos grupos da etapa superior, formando uma hierarquia". Como exemplos, temos os algoritmos de ligação simples e ligação completa. Em um algoritmo aglomerativo [Carvalho 01], cada objeto inicia formando um grupo com ele mesmo e em cada etapa os mais próximos são combinados para formar um novo grupo até reunir todos objetos em um único grupo.

Soh [Soh & Tsatsoulis 99] faz uso do algoritmo COBWEB/3 para agrupar regiões por características espectrais, espaciais e de textura de imagens de satélites. Nesse caso, a mineração de imagens foi usada como um instrumento auxiliar ao processo de visão computacional. O funcionamento do COBWEB/3 consiste na formação de conceitos probabilísticos, no qual uma instância pertence a um conceito com um certo grau de certeza. Em resumo, a cada iteração, um algoritmo desse tipo agrupa as instâncias em conceitos, sumariza-as e organiza-as hierarquicamente. Em [Zhang et al. 01], são citados outros exemplos que envolvem agrupamentos, mas a maioria destes estão envolvidos em estágios iniciais do processo de mineração em que após a execução do algoritmo, normalmente é necessária a intervenção humana para rotulação dos grupos gerados.

5.3 DESCOBERTA DE REGRAS DE ASSOCIAÇÃO

Em [Ordonez & Omiecinski 99] é apresentado um sistema para busca de regras de associação em imagens coloridas, como segue:

- a) extração de características: segmentação das imagens em regiões chamadas de *blobs* [Carston et. al 99], que inclui informações de cor, textura, forma, área e posição;
- b) identificação de objetos e criação de registros: comparam-se os *blobs* de cada imagem com as demais, associando a cada *blob* uma identificação - ao final, um mesmo *blob* poderá estar presente em outras imagens;
- c) criação de imagens auxiliares: para visualização dos resultados, comparando os *blobs* com as imagens originais;
- d) mineração de imagens: executa a descoberta de regras de associação, baseado em registros que possuem os *blobs* e suas respectivas características.

Em [Djeraba 01], o autor propõe um sistema semelhante para descoberta de relacionamentos escondidos em características de imagens agrupadas em dicionários. Combinando soluções de SRIC e aprendizado de máquina, a extração de características combina o que o autor chama de agrupamento simbólico e a descoberta de relacionamentos propriamente dita. A primeira etapa se baseia em uma fase de pré-processamento, na qual é feita a montagem de um dicionário de características, que é o agrupamento das características mais semelhantes, de maneira semelhante ao *blob* citado no algoritmo anterior. Na seqüência, para cada grupo de características de uma imagem é associada uma representação simbólica baseada no dicionário para simplificar a fase de descoberta de relacionamentos.

Em [Zaïane & Han 00, Zaïane et al. 00], é usado o algoritmo MaxOcurr, modificação do algoritmo *Apriori*, para extrair regras com itens repetidos (*recurring items*) a partir de uma base de dados espaciais. Uma das observações é que dados visuais possuem propriedades específicas do domínio de imagens, de modo que algumas características visuais podem estar repetidas em uma mesma imagem e essas ocorrências podem ser consideradas relevantes. O processo proposto por Zaïane pode ser resumido nos passos a seguir:

- a) segmentação de regiões baseada em cor e textura, dando origem a áreas que compartilham algumas características, tais como massa, variância e centróide;
- b) identificação de relações espaciais nas áreas extraídas, tais como *disjoint*, *inside*, *contains*, *equals*, *meets*, *covered by*, *covers* e *overlaps*, usando diferentes resoluções da imagem (*resolution refinement*), para melhoria de performance;
- c) aplicação do algoritmo de extração de relações.

5.4 LÓGICA FUZZY

A teoria *fuzzy* [Zadeh 65, Masulli & Schenone 99, Pal 92, Steimann 97], define uma função que indica se um elemento pertence ou não a um conjunto, porém ao invés de ser binária, indica o quanto o elemento pertence, numa escala de 0 a 1. Juntamente com essa função, são definidos todos os operadores que atuam no conjunto, enfim, toda a base lógica que precisa existir para sua utilização prática. Essa teoria tem grande uso em tarefas de classificação e agrupamento principalmente em problemas na área de medicina e também em algoritmos de segmentação.

5.5 MÉTODO DE BAYES E REDES BAYESIANAS

A regra de Bayes é um conceito estatístico [Downing & Clark 00] que mostra como achar a probabilidade condicional $P(B|A)$ uma vez conhecida a probabilidade de $P(A|B)$. O método Naïve Bayesian [Witten & Frank 00] consiste em assumir, baseado na regra de Bayes, que todas as variáveis envolvidas são independentes, o que pode ser considerado algo “ingênuo”, pois na prática nem sempre é possível considerar essa premissa. Os autores destacam que por mais simples que seja essa

estratégia, ela funciona bem com conjuntos de dados que passaram por procedimentos de filtragem de campos redundantes. Também com grande aceitação, redes Bayesianas têm sido usadas principalmente devido a seu caráter estatístico e por ser, principalmente, um modelo para representação de conhecimento, construção e interpretação, como visto em [Heckerman 96]. A utilização é semelhante à de uma rede neural, com a diferença que sua representação de nós e arcos é mais fácil para a modelagem de uma base de conhecimento.

Uma aplicação direta do método de Bayes para mineração de imagens é o sistema GeoBrowse [Koperski & Marchisio 00], que provê uma interface para consultas por similaridade e um módulo de análise de clusterização de imagens de satélites. Para cada imagem, são extraídos três tipos de vetores de características, conforme a classificação do dado: nível de pixel (cor, textura); nível de região (contorno, forma, tamanho, dentre outros atributos), e; nível de faixa (*tile*) (semelhante a região, porém em relação à imagem inteira). No GeoBrowse, contudo, o uso do método Bayesiano está no aprendizado da descrição de vetores de características baseado na realimentação do usuário para classificação de tipos coberturas de terra. O usuário treina o modelo indicando as instâncias positivas e negativas de grupos de vetores de características do nível de pixel. Assim, devido ao caráter de independência dos atributos do algoritmo Naïve Bayesian, a pesquisa de imagens será feita usando um agregado de todos os *pixels* da imagem.

Vailaya demonstra um exemplo de mineração de imagens envolvendo redes Bayesianas, onde há a classificação hierárquica de imagens fotográficas diversas [Vailaya et al. 96]. O objetivo é treinar o classificador para avaliar as imagens segundo conceitos de alto nível, como fotografias de cenas internas/externas, cidade/paisagem e pôr-do-sol/floresta/montanha.

5.6 REGRESSÃO

Para [Witten & Frank 00], modelos de regressão são usados quando a classe a ser descoberta é numérica, bem como todos os atributos envolvidos. O objetivo é expressar em uma fórmula a combinação dos atributos e pesos aplicados a cada atributo, sendo que cada peso será obtido através dos dados de treinamento. Aplicando-se uma instância de valor não conhecido na fórmula obtida, obtém-se o valor da classe, permitindo prever o valor da nova instância.

Os autores de [Kakimoto et al. 00] utilizam o algoritmo *Logical Regression Analysis* (LRA) para identificar áreas ativas do cérebro executando determinadas funções. Para isso, os autores usaram uma combinação de regressão para extração de fórmulas que descrevem o comportamento do cérebro com um algoritmo de geração de regras que permitem uma interpretação melhor do resultado. Nesse mesmo trabalho, como primeira fase do algoritmo LRA, é aplicada a regressão não-paramétrica ao invés da regressão linear. O que motiva o uso desse tipo de abordagem é que o número de exemplos é menor que o número de variáveis independentes, além do fato que devido ao relacionamento forte entre os *pixels*, não é possível assumir que se trata de um caso de variáveis realmente independentes.

5.7 REDES NEURAIIS

Para [Azuaje et al. 99], uma rede neural é um processador paralelo que possui a capacidade natural de armazenar conhecimento através de experiência e aplicá-lo, simulando o processo de aprendizado do cérebro através de conexões neuronais, que são os pesos sinápticos que armazenam o conhecimento. Um dos princípios do aprendizado desse tipo de algoritmo está na retropropagação de erros que consiste na “propagação dos erros dos neurônios da camada de saída em direção à entrada, possibilitando a alteração das sinapses para o aprendizado” [Carvalho 01].

5.8 INSTANCE-BASED LEARNING RACIOCÍNIO BASEADO EM CASOS

Algoritmos de *Instance-based learning*, como o IBk [Aha & Kibler 91], também conhecido como “vizinho mais próximo” (*K-Nearest Neighbor*), são baseados no aprendizado de uma função de distância que determina o quanto uma instância da base de testes está próxima a uma instância da base de treinamento. Para isso, o sistema assume que todas as instâncias correspondem a pontos em um espaço n -dimensional. Segundo [Mitchel 97], o aprendizado nessa abordagem consiste simplesmente em armazenar, de maneira consistente, os dados de treinamento e compará-los com a instância usada como parâmetro numa consulta. Mais adiante, na Seção 8, esse tipo de algoritmo será usado nos experimentos de Mineração de Imagens.

Uma outra técnica usada para descoberta de conhecimento é a de Raciocínio Baseado em Casos que é uma “abordagem para solução de problemas baseada em soluções similares no passado” [Althoff et al 98], isto é, também utiliza similaridade de instâncias mas não representa as mesmas num espaço Euclidiano. Bastante usada na medicina, essa técnica usa noções de similaridade entre problemas para recuperação de exemplos em uma base de casos. Um caso é a descrição do problema e sua solução. Seu uso em Mineração de Dados é evidente, tendo sido usado para vários tipos de problemas, como classificação, recuperação de imagens de radiologia e interpretação de imagens do miocárdio [Althoff et al 98].

5.9 PROGRAMAÇÃO DE LÓGICA INDUTIVA

Uma outra técnica, fortemente relacionada com aprendizado de máquinas é a Programação em Lógica de Indução. Em [Dzeroski 96], o autor a define como uma técnica de “aprendizado de máquina em uma linguagem de lógica de primeira ordem”. Sua aplicação em KDD é vista na descoberta de regras que envolvem atributo-valor, descoberta de relacionamentos em uma base de dados, aprendizado de múltiplos predicados e tarefas de classificação. [Siromoney et al. 00] utilizam

uma ferramenta chamada Progol para aprendizado de conceitos, junto com uma base de conhecimento para gerar hipóteses. Nesse caso, os dados constituem uma base de imagens médicas e a mineração de dados é feita com o objetivo de discriminar tipos de anomalias e relações entre regiões das imagens.

5.10 COMPUTAÇÃO EVOLUCIONÁRIA E ALGORITMOS GENÉTICOS (AG)

A computação evolucionária utiliza a teoria Darwiniana como um processo adaptativo de otimização, englobando, além de AG, programação evolucionária (PE), estratégias evolucionárias (EE), programação genética (PG) e sistemas classificadores (SC). Os AG são mais difundidos e são utilizados principalmente para otimização, devido a sua eficácia em realizar busca global [Peña & Sipper 99, Tanomaru 95]. Em AG, uma população de possíveis soluções para o problema em questão evolui de acordo com operadores probabilísticos concebidos a partir de metáforas biológicas, de modo que há uma tendência que, na média, os indivíduos representem soluções cada vez melhores à medida que o processo evolutivo continua. Essas metáforas biológicas são conceitos como cromossomos, evolução, reprodução com *cross-over* e mutação. Essa técnica pode ser utilizada em classificação, análise de agrupamentos e aprendizado de redes neurais.

[Reyes & Sipper 99] destacam vários trabalhos que fazem uso de computação evolucionária com imagens médicas, em alguns casos, aplicada para otimização de parâmetros de outros algoritmos e em outros casos diretamente para extração de informações úteis das imagens. A gama de exemplos envolve radiografia de tórax, tomografia computadorizada, imagens de ultra-som, mamografia, ressonância magnética, dentre outros.

5.11 ROUGH SETS

Os autores de [Komorowski et al. 98] definem que *rough sets* são utilizados para "sintetizar definições de conceitos aproximados de dados". Em linhas gerais, os algoritmos de *rough sets* buscam os atributos que melhor descrevem ou tornam indiscerníveis um conjunto de tuplas, dando origem a uma função que indica se uma tupla pertence ou não a um conjunto.

5.12 OUTROS ALGORITMOS.

Embora baseado em um princípio físico, o resfriamento simulado [Carvalho 01] é uma alternativa bastante semelhante a um algoritmo genético, sendo ambos usados para otimização global. Um outro algoritmo, é o algoritmo elástico, útil na análise de agrupamentos, que pode ser descrito como um "procedimento iterativo que parte de um círculo que é gradualmente alongado não uniformemente até que passe suficientemente próximo de todos os pontos definindo uma trajetória sobre os mesmos" [Carvalho 01].

6 ARQUITETURA DE UM SISTEMA DE MINERAÇÃO DE IMAGENS

Com base nas estratégias, técnicas e algoritmos apresentados é possível estabelecer uma classificação dos Sistemas de Mineração de Imagens. Nessa Seção, serão descritas as abordagens e estruturas mais comuns para a construção de um sistema de mineração de imagens, isto é arquiteturas que demonstram uma maneira híbrida e eficaz de combinar as abordagens *bottom-up* (as características guiam o processo de interpretação) e *top-down* (o modelo guia a extração de características e a interpretação).

6.1 CLASSIFICAÇÃO QUANTO À ABORDAGEM: ORIENTADA A MODELO X ORIENTADA A DADOS

Uma das principais questões que envolvem a escolha de uma arquitetura se refere a como, a partir dos dados, chegar a um modelo e, a partir do modelo, inferir o comportamento dos dados (vide Figura 4). Em [Weschler 90], o autor define as fases de um Sistema de Visão Computacional de maneira semelhante a [Jain et al. 95], porém destaca as fases de extração de características (*bottom-up*) e de fase de predição e invocação do modelo (*top-down*). Na metodologia orientada a dados ou *bottom-up*, baseada nas características de baixo nível da imagem, o processamento começa na própria imagem e termina no reconhecimento do objeto. Conforme [Sonka et al. 99], na fase de pré-processamento, a imagem sofre transformações que irão salientar suas características, que serão detectadas e agrupadas na fase de segmentação. Por fim, as regiões obtidas irão passar por técnicas de reconhecimento de padrões, isto é, serão comparadas com o modelo. Por outro lado, a abordagem orientada a modelos fundamenta-se na geração e teste de hipóteses. Para [Sonka et al. 99], o princípio dessa abordagem é “a construção de um modelo interno e sua verificação, ou seja, é um processamento orientado a

objetivo". O mecanismo consiste no modelo orientar a extração das características de baixo nível de uma imagem e validá-las na mesma imagem ou em outras [Brooks 81, Fisher 89].

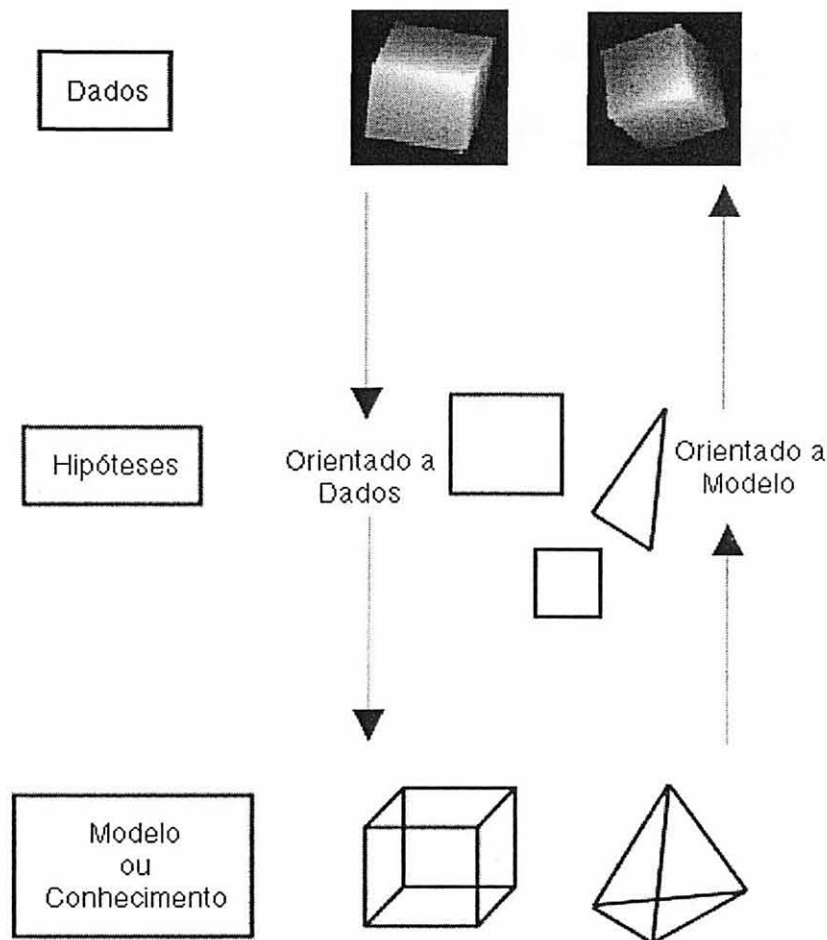


FIGURA 4 - ORIENTADO A DADOS X ORIENTADO A MODELO

Na prática, um sistema de visão computacional ideal combina as duas arquiteturas, no que [Sonka et al. 99] denomina de estratégia combinada. Uma solução híbrida é mais robusta pois combina a capacidade da estratégia orientada a dados de encontrar regiões e formas sem intervenção humana com a capacidade que a estratégia orientada a modelo tem para guiar os processamentos de baixo nível. No caso dos SRIC, a abordagem combinada tem várias aplicações. Na fase de pesquisa, o modelo seria o guia da construção da consulta e, na etapa de validação,

poderia fazer o papel que um usuário faria na fase de realimentação de relevância. Também é na arquitetura híbrida que a mineração de dados vem dar uma de suas maiores contribuições. Numa primeira fase, a mineração de imagens irá auxiliar na construção do modelo, inferindo regras, associações e outros tipos de estruturas a partir das características das imagens. Na seqüência, o modelo gerado será testado para detectar objetos ou auxiliar em outras atividades de processamento de imagens e de visão computacional.

6.2 CLASSIFICAÇÃO QUANTO À MODULARIDADE: ORIENTADO À FUNÇÃO X INFORMAÇÃO

Em [Zhang et al. 01], classifica-se os atuais sistemas de mineração de dados em orientado à função e orientado a dados, conforme a figura a seguir:

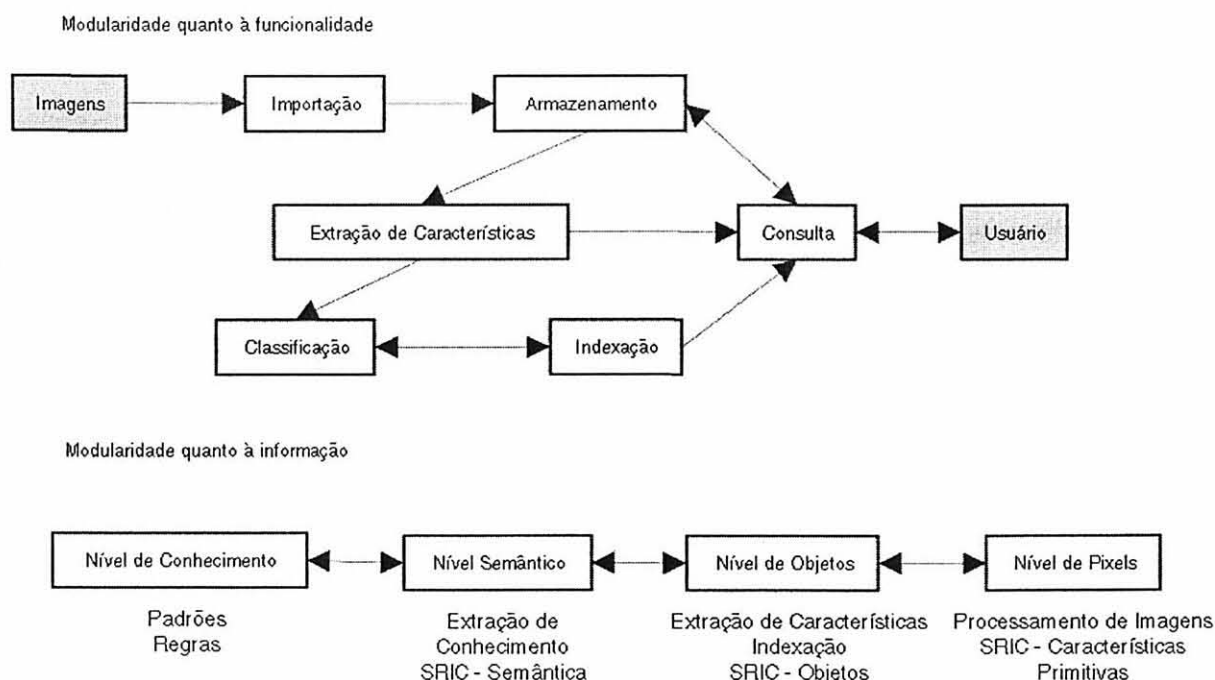


FIGURA 5 - CLASSIFICAÇÃO QUANTO À MODULARIDADE [Zhang et al. 01]

Na primeira categoria, encontram-se sistemas como os de [Datu & Seidel 00, Zaiane et al. 98], que são divididos em componentes, cada um responsável por uma atividade do processo de mineração. Na segunda categoria, o foco não é a modularização e sim a estrutura dos dados. Os dados são divididos em vários níveis de informação: nível de pixel, nível de objeto, nível semântico e nível de padrões e conhecimento. Os SRIC descritos em [Bellon et al. 02, Gorga et al. 02] são exemplos da classificação orientada à informação, uma vez que é projetado um sistema que utiliza diferentes níveis de conhecimento e também prevê o uso de Mineração de Dados.

6.3 CLASSIFICAÇÃO QUANTO AO TIPO DE APLICAÇÃO

Como aplicação de processamento de imagens, a mineração vem auxiliar os algoritmos de baixo nível, como a segmentação. Os autores de [Soh & Tsatsoulis 99] adaptam técnicas de mineração de dados para analisar e segmentar imagens de satélites. Primeiramente, utilizam-se processos de segmentação e análise como limiarização e geração de valores de textura. Essas características obtidas, junto com a imagem original, são então tratadas como uma base de dados em que vão ser aplicados algoritmos de agrupamento, representando os objetos identificados. Aqui, uma única imagem é minerada, mas a quantidade de características é suficiente para a aplicação de técnicas de mineração.

Quando usada para identificação e classificação de objetos, a mineração de imagens atua como uma ferramenta de visão computacional. O sistema SKYCAT [Fayyad et al. 96a] é responsável pela automatização e catálogo de imagens astronômicas onde um algoritmo de classificação é treinado com essa base para poder aumentar o catálogo de objetos e validar novos modelos. Em [Smyth et al. 96], os autores também usam uma abordagem semelhante para identificação de vulcões em Vênus a partir de imagens de radar.

Uma área pouco explorada é o uso da Mineração de Imagens para auxiliar SRIC. Em [Eakins 02], o autor afirma que a recuperação de imagens envolvendo

objetos e cenas complexos baseada em semântica requer um sistema adaptativo capaz de aprender com experiência, além de salientar que é impraticável para quem mantém um SRIC modificar o modelo de objetos a cada novo exemplo encontrado, ou seja, o sistema tem que ser capaz de evoluir sozinho. Em [Smeulders et al. 00], é destacada a importância do componente de aprendizado para melhorar a eficácia dos SRIC, uma vez que essa solução ganha cada vez mais atenção com o aumento do tamanho das bases de dados e da performance das máquinas.

7 PROJETO SRIC3D

O SRIC3D é o projeto de um Sistema de Recuperação de Imagens por Conteúdo, atualmente em desenvolvimento pelo Grupo IMAGO. O diferencial deste projeto está no uso de imagens de profundidade (*range images*) como base de dados e também na utilização de Mineração de Imagens como auxiliar ao sistema de busca. Sua arquitetura também envolve a reconstrução de modelos 3D a partir das vistas de objetos de maneira integrada aos módulos de recuperação de imagens e de mineração. O uso combinado de imagens de profundidade e modelos 3D abre possibilidade para vários tipos de aplicações, tais como: museu virtual, engenharia reversa, sensoriamento remoto e medicina. O projeto está desenvolvendo a fase de extração de características e espera-se, como resultado, a determinação da aplicabilidade da Mineração de Imagens como uma ferramenta de Visão Computacional e de SRIC.

7.1 TRABALHOS RELACIONADOS

Nesse tópico serão vistos alguns trabalhos que envolvem a pesquisa de objetos ou cenas a partir de informações 3D, sejam essas informações mapeadas em matrizes (imagens) ou modelos. Todas essas abordagens terão como saída várias características que servirão de apoio para o projeto SRIC3D, em especial, para o módulo de mineração de imagens.

7.1.1 Imagens de profundidade

Um dos focos da Visão Computacional é a exploração de imagens de profundidade, que possuem informação 3D explícita. De acordo com [Besl 88], imagens de

profundidade são grandes matrizes tal que o valor de cada posição é dado pelas medidas do sistema de coordenadas de um sensor até os pontos da superfície de um objeto. Nesse contexto, é necessário o desenvolvimento de algoritmos de segmentação específicos para imagens de profundidade, tais como:

- a) segmentação usando superfícies planares [Bellon et al. 99a];
- b) segmentação por bordas [Bellon et al. 99b, Bellon & Silva 02];
- c) segmentação por mapa das curvaturas Média e Gaussiana das superfícies do objeto [Bellon et al. 01], e;
- d) extração de características [Gotardo 02, Nascimento 02, Silva 03].

Fora do contexto de reconhecimento de objetos, porém mais perto dos SRIC, constata-se que poucos trabalhos fazem uso de imagens de profundidade. Usando uma proposta semelhante a do projeto SRIC3D, [Soodamani & Liu 00] descrevem um sistema para reconhecimento de objetos cuja arquitetura faz uso de características de mais alto nível de imagens de profundidade, tais como as curvaturas principais e momentos centrais invariantes. Além das características já previstas para o SRIC3D, destaca-se a ênfase que o sistema de Soodamani tem no aprendizado para atualização do modelo para reconhecimento.

7.1.2 Modelos tridimensionais.

Outra abordagem envolve sistemas de recuperação e indexação de modelos 3D [Shih et al. 02]. Os exemplos demonstrados usam a representação VRML² ou *Virtual Reality Modeling Language*, que é uma linguagem para modelagem de objetos 3D e realidade virtual, definindo um formato de arquivo que integra gráficos 3D e multimídia [Carey et al. 97]. A escolha do formato, linguagem ou padrão para representação 3D é importante para a modelagem do banco de dados, mas não implica que o formato VRML será o escolhido. Em Mahmoudi & Dadoudi 02, os autores usam como características de indexação as vistas dos objetos (3 principais e 4 secundárias), representadas pelo seu contorno, utilizando o descritor CSS (*Curvature Scale Space*) que, por sua vez, é indexado utilizando a estrutura *M-Tree*. Em [Croquette et al. 02], os autores também trabalham na indexação de modelos VRML, mas combinando a indexação de dois descritores: um local, baseado em

forma, e; um global, baseado em um histograma da distribuição da distância entre pontos aleatórios de superfícies. Além dos tipos de descritores, o interessante dessa solução é que se faz uso de mais de um tipo de índice para melhorar a eficiência do SRIC. Por sua vez, o trabalho de Gudivada [Gudivada & Jung 95] utiliza um grafo, SOG (*Spatial Orientation Graph*), para representar os objetos. A pesquisa é feita comparando-se o SOG da imagem exemplo com os grafos SOG das imagens armazenadas em um banco de dados espaciais.

Com base nos trabalhos descritos acima e nas Seções anteriores, observa-se que as soluções de Mineração de Imagens não trabalham com as características básicas isoladamente, mas em conjunto com outras representações de forma, de relacionamento entre regiões e de mecanismos de indexação. Um dos desafios é descobrir quais as características mais significativas e de que maneiras usá-las adequadamente para obter um mecanismo de pesquisa de imagens eficiente. Essa é mais uma razão da inclusão da Mineração de Imagens como um dos principais componentes do SRIC3D, como será visto na Seção a seguir.

7.2 ARQUITETURA DO SISTEMA

Na Seção 3, foi detalhada uma adaptação do processo de Mineração de Dados para a realidade da Mineração de Imagens. Além de seguir essa proposta, o SRIC3D também inclui outros passos específicos para o tratamento de imagens de profundidade e a reconstrução de modelos 3D. Uma visão geral da arquitetura do SRIC3D pode ser vista na Figura 6:

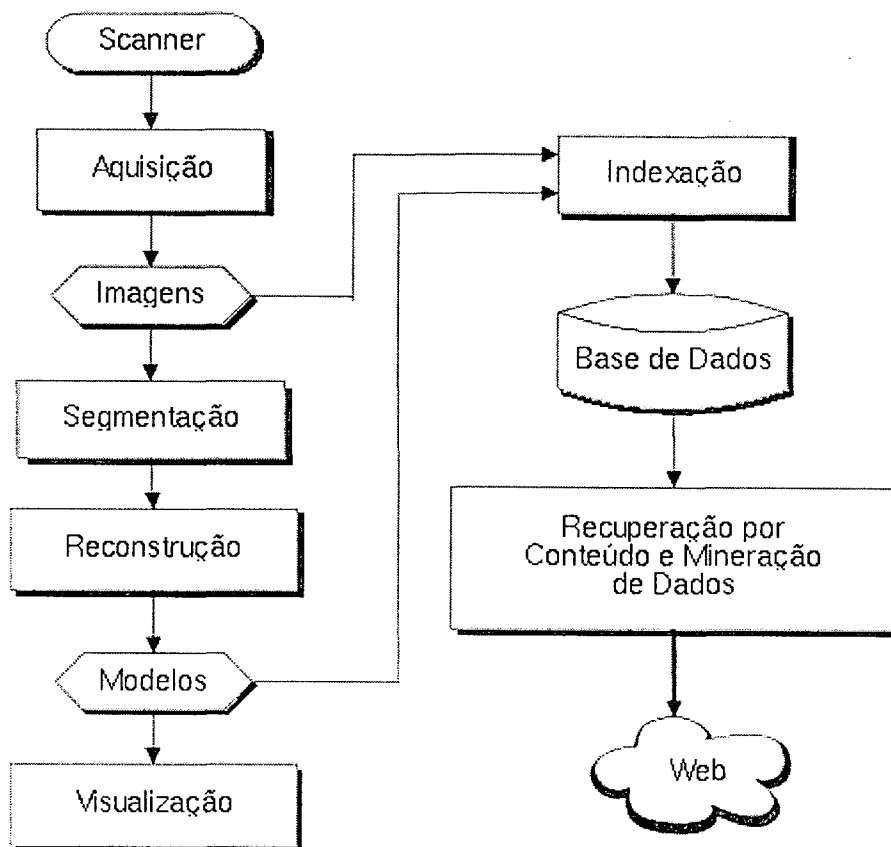


FIGURA 6 - ARQUITETURA DO SRIC3D

Primeiramente, será necessária a aquisição das imagens, o que caracteriza a fase de **obtenção**. O SRIC3D usará tanto imagens obtidas de *scanner* (no caso, um *MODELA MDX-15 touch-probe*) quanto de bases de dados de imagens de profundidade disponíveis na Internet. Inicialmente, contudo, será usada para testes apenas a base *OSU (MSU/WSU) Range Image Database*⁵, que já possui cerca de 700 objetos (sendo que cada objeto possui várias vistas). Como os algoritmos de segmentação implementados pelo Grupo IMAGO são baseado nas bibliotecas da ferramenta KHOROS⁶, as imagens serão convertidas do formato original para KDF (*KHOROS Data File*) e separadas em bandas (eixos X, Y e Z).

⁵ <http://sampl.eng.ohio-state.edu/~sampl/data/3DDB/RID/index.htm>

⁶ <http://www.koral.com>

Uma vez convertida, a imagem será o ponto de partida para um conjunto de etapas de **pré-processamento** (melhoramento da imagem) e **geração de características** e, que é ilustrada na Figura 7. Inicialmente serão obtidas características globais da imagem através de vários procedimentos, tais como:

- a) isolamento das bordas de profundidade com base apenas na informação da ordenada Z, para auxílio do cálculo das normais;
- b) cálculo das normais por métodos robustos [Silva 03];
- c) análise da matriz de co-ocorrência, gerando 14 características: segundo momento angular, contraste, correlação, variância, homogeneidade, soma da média, soma da variância, soma da entropia, entropia, diferença da variância, diferença da entropia, informação de medidas de correlação e coeficiente máximo de correlação [Haralick et al. 95];
- d) afinamento de bordas [Silva 03, Silva 02], através de uma operação morfológica que reduz a espessura das linhas para 1 *pixel* de largura (esqueleto), utilizando a transformada “hit-or-miss”;
- e) poda [Gonzales & Woods 92], para eliminação de *pixels* redundantes ou “parasitas”, resultantes dos processos de afinamento e esqueletonização, e;
- f) a partir da poda e do afinamento, a borda recém calculada será vetorizada ou poligonalizada, gerando como características as principais arestas do polígono que formam a borda do objeto.

Na seqüência, o algoritmo *K-Means* segmentará a imagem em regiões, considerando como atributos o vetor normal e um algoritmo de rotulação, que irá identificar cada região. A partir dessa identificação, o sistema irá extrair, para cada região, várias características locais, conforme os passos a seguir:

- a) cálculo das bordas de cada região;
- b) poligonização da borda por região;
- c) cálculo do centro de massa, baseado no mínimo retângulo envolvente de cada região;
- d) descoberta de todas as regiões que são vizinhas à região sendo analisada, já prevendo o uso de estruturas de dados espaciais, e;
- e) obtenção da informação de ângulo e do vetor normal de cada ponto de interseção entre as regiões.

Além das regiões, textura, vizinhança, normais, ângulo em relação a vetor referência e informações espaciais, outras características também serão utilizadas, e.g., curvaturas Média e Gaussiana [Besl 88], características obtidas por meio de *wavelet* [Liank & Kuo 99] e momentos invariantes [Croquette et al. 02]. O objetivo é utilizar o maior número de atributos para identificar relacionamentos e avaliar a importância e validade de cada característica, dando prioridade a atributos invariantes à rotação, translação e escala.

A fase de **modelagem de dados**, que ocorre em paralelo com as demais fases anteriores à mineração de dados, consiste em procurar a melhor maneira de representar as características encontradas. No SRIC3D, está sendo avaliada a possibilidade de usar não somente dados textuais e numéricos, como também dados espaciais, aproveitando algumas ferramentas próprias do gerenciador de banco de dados escolhido, o PostgreSQL. É na fase de modelagem que serão definidas as estruturas e tabelas para armazenamento do modelo 3D, tabelas e campos para gravação das características e a estrutura para representação do conhecimento obtido na mineração de imagens. Também com base nessa modelagem, será feita a escolha da estrutura de índices apropriada para consulta dos objetos armazenados, isto é, a fase de **indexação**. O SRIC3D prevê o uso de algumas extensões do PostgreSQL, como a que permite a escolha de diferentes esquemas de indexação, inclusive *R-Tree*.

O próximo passo é a execução do módulo de **mineração de dados**. Em uma primeira etapa, o foco será a descoberta de relações, com o objetivo de analisar a validade das características e de seus relacionamentos. Para auxiliar essa exploração, serão utilizados algoritmos baseados em bibliotecas de código aberto,

tais como Rosetta [Øhrn et al. 98] e WEKA [Garner 95]. As conclusões obtidas desse estudo servirão para avaliar a aplicabilidade da mineração de dados nas fases de recuperação de imagens e de reconstrução 3D de imagens a partir de várias vistas de imagens de profundidade. Posteriormente, outras estratégias de mineração também poderão ser aplicadas, principalmente a classificação. Esses fatos também ressaltam a importância das fases de **refinamento** para o armazenamento das regras, e da fase de **aplicação e teste**, para a validação do SRIC3D.

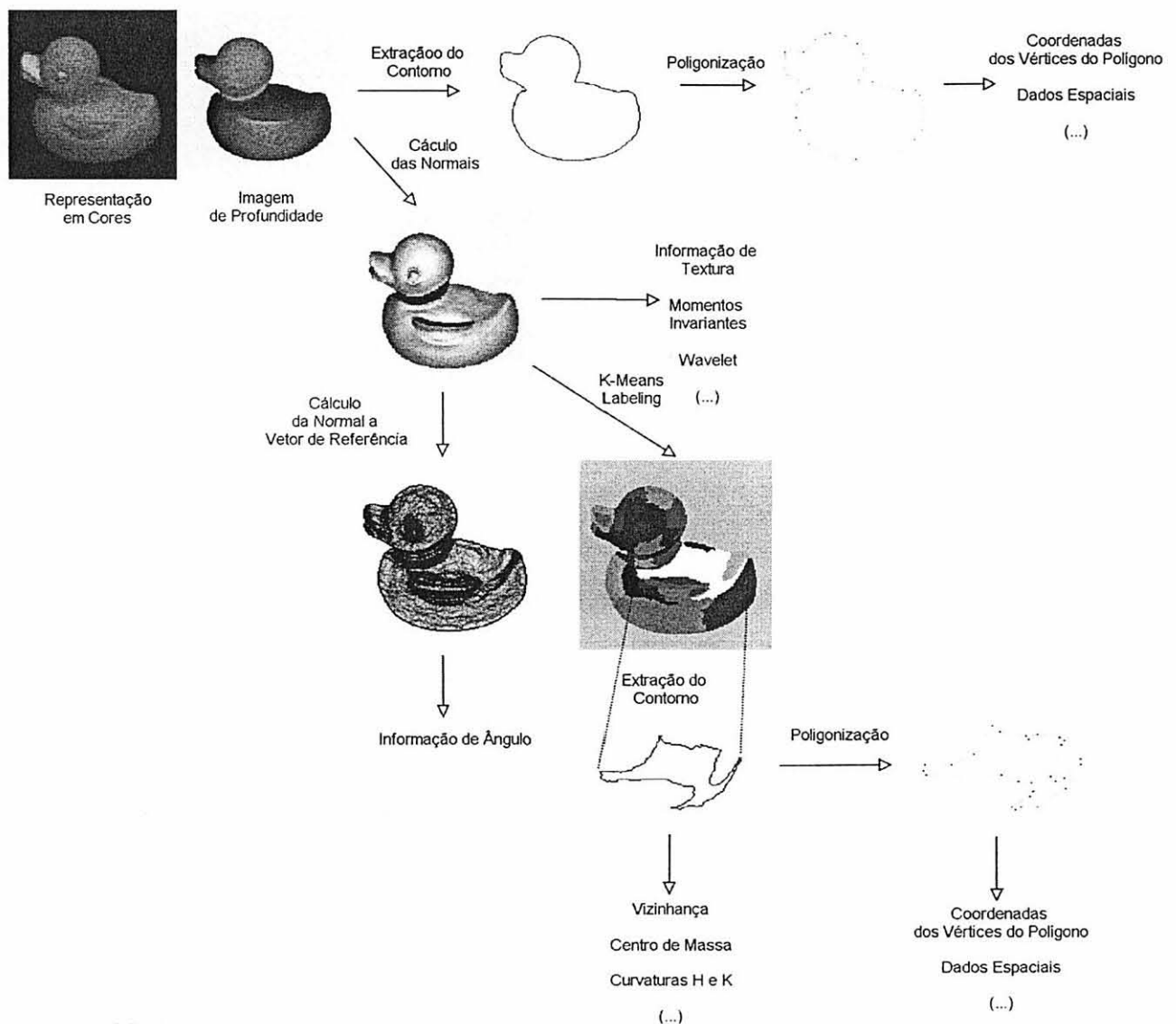


FIGURA 7 - ATIVIDADE DE EXTRAÇÃO DE CARACTERÍSTICAS

O módulo de **reconstrução 3D e visualização** terá como objetivo a geração de modelos 3D a partir das diversas vistas das imagens [Silva 03]. Apesar de utilizar

características geradas nas fases discutidas nos parágrafos anteriores, essa atividade baseia-se essencialmente em heurísticas e técnicas que trabalham com aspectos de nível mais baixo da imagem e outras abordagens de reconhecimento de objetos. Reitera-se a importância das relações e classificações oriundas da mineração de imagens como possível auxiliar da fase de reconstrução 3D.

Por fim, o SRIC3D conterá um módulo de **recuperação de imagens por conteúdo**. Não se trata de um subsistema isolado, pois irá utilizar a mesma estrutura de indexação, modelagem e características comuns ao módulo de mineração de imagens. O desafio consiste em procurar a melhor maneira de disponibilizar uma interface para o usuário fazer consultas baseadas nas propriedades das imagens de profundidade. Com base nos resultados experimentais, será determinada uma interface de consulta para o usuário compatível com os tipos de dados armazenados e das características de alto nível que se deseja obter. Isso implica que, além dos parâmetros tradicionais de pesquisa (imagem-exemplo e características de baixo nível e nível intermediário), será preciso definir como utilizar as regras extraídas e as relações entre objetos obtidas na mineração como critérios de busca.

Levando em consideração as etapas do sistema e a classificação de arquiteturas descrita na Seção 6, conclui-se que o SRIC3D possui uma arquitetura híbrida quanto à abordagem, pois irá combinar tanto a metodologia *bottom-up* para a construção do modelo por meio da mineração de dados, quanto *top-down*, para aplicação do modelo para a recuperação das imagens. O resultado é um sistema modularizado, orientado a função, com etapas de aquisição, pré-processamento, descoberta de conhecimento e execução de consultas bem definidas. Uma das principais contribuições do SRIC3D está no uso da Mineração de Imagens como um módulo auxiliar a um SRIC, para validar o quanto as características de alto nível podem tornar a consulta de imagens por conteúdo mais eficiente.

8 FERRAMENTA DE MINERAÇÃO DE IMAGENS

Tomando por base uma solução genérica de mineração de imagens e sua aplicação em um sistema de recuperação de imagens por conteúdo, foi desenvolvida uma ferramenta híbrida, dividida em módulos, de modo que o reaproveitamento de cada componente possa ser feito independentemente dos demais. A solução proposta não está presa apenas à plataforma adotada (no caso Java), podendo ser adaptada facilmente para qualquer outro tipo de linguagem de programação ou ambiente de desenvolvimento, principalmente porque foi adotado o paradigma de orientação a objetos para separação das funcionalidades.

Primeiramente, será visto na seção 8.1 como os componentes arquiteturais estão organizados e de que modo eles interagem no processo de mineração de imagens. Na seção 8.2, cada componente será descrito com mais detalhes quanto a sua estrutura interna (atributos e métodos). Na seção 8.3, será descrito como os componentes do módulo de mineração e consulta interagem internamente. Finalmente, na seção 8.4, a arquitetura proposta será aplicada no domínio de imagens de profundidade, já apresentando uma solução de recuperação de imagens por conteúdo na qual o usuário final pode interagir.

8.1 VISÃO ARQUITETURAL

Conforme a Figura 8, o sistema está dividido em camadas e módulos. A implementação é fortemente baseada em Java, tornando-se natural o uso das APIs da própria linguagem, tais como, JSP/Servlets⁸ para aplicações Web ou Swing⁹ para desktop.

⁷ API – Application Programming Interface – bibliotecas de programação

⁸ JSP/Servlets – API Java para programação de sistemas via Internet

⁹ Swing – API Java para aplicativos desktop, semelhante ao Visual Basic ou Delphi

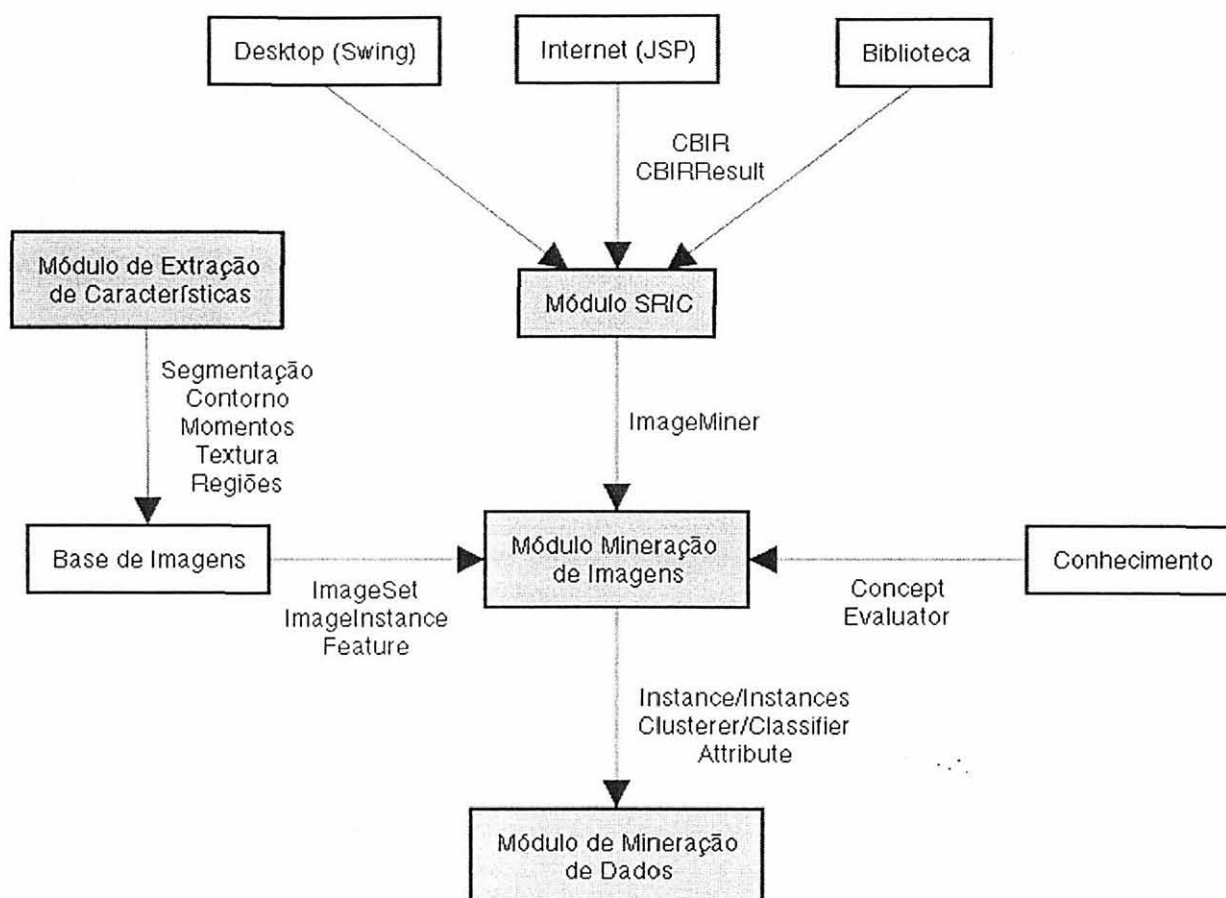


FIGURA 8 – ARQUITETURA DA FERRAMENTA DE MINERAÇÃO DE IMAGENS

Também foi utilizado JDBC¹⁰ para acesso transparente a banco de dados, além da biblioteca de Mineração de Dados, WEKA. Através do uso de UML¹¹ (*Unified Modeling Language*) para documentação do sistema e de padrões de Engenharia de Software para modelagem, torna-se possível converter a solução para qualquer linguagem de programação, bastando adaptar a implementação para outras bibliotecas de mineração de dados ou de interface gráfica.

O módulo de Mineração de Imagens, foco deste trabalho, é essencialmente uma biblioteca que serve de ligação entre o banco de dados, o módulo de consulta e a biblioteca de Mineração de Dados. Dessa maneira, a camada de interação com o usuário fica totalmente independente de implementação, permitindo que o sistema

¹⁰ JDBC – Java Database Connectivity, API para acesso a base de dados, no caso PostgreSQL

¹¹ UML – Linguagem de Modelagem Unificada – para confecção de diagramas usando notação orientada a objetos

possa ser desenvolvido para *desktop* (aplicativo rodando em apenas uma máquina) ou para Web (no qual o usuário, a partir do *browser*, submete consultas a um servidor Web de aplicações).

8.1.1 Módulo de Mineração de Dados

Esse módulo é uma biblioteca independente, que possui diversos algoritmos de classificação, agrupamento e de descoberta de regras de associação. O módulo de Mineração de Imagens irá encapsular os métodos e a configuração dos algoritmos de mineração providos pela biblioteca WEKA, além de incluir novas funcionalidades, tais como: permitir sua aplicação em cada imagem no banco de dados para posterior avaliação; persistência do modelo; múltiplos conceitos ou classes por registro.

A biblioteca WEKA aplica os algoritmos de mineração sobre um conjunto de dados representado pela classe *Instances*, que corresponde a uma tabela de um banco de dados. Cada registro ou instância é representado por um objeto da classe *Instance*. Essas tabelas podem ser usadas para treinamento e avaliação de um ou mais algoritmos através de diferentes estratégias, tais como:

- a) utilizando um grupo distinto de instâncias para aprendizado e outro para teste;
- b) particionamento, que separa a base de dados em dois grupos, sendo uma porcentagem voltada para o aprendizado do algoritmo, e o restante para avaliação, e;
- c) *cross-validation*, que separa as instâncias em subdivisões menores que são avaliadas diversas vezes, na qual a cada iteração é utilizada uma combinação diferente de instâncias de treinamento e teste.

Cada campo ou propriedade de uma instância é um objeto da classe *Attribute*. Um desses atributos poderá ser indicado como aquele que representa a classe a qual a instância pertence, isto é, será utilizado para o aprendizado supervisionado dos algoritmos de classificação. Um atributo pode ser numérico (a

maioria das características das imagens são valores numéricos), data/hora (útil para bancos de dados tradicionais) ou nominal. Atributos nominais, como descritos na Seção 3.5, são *strings* válidas dentro de um conjunto de valores possíveis. A classe *object_type* (tipo de objeto) utilizada mais adiante no aplicativo de Mineração de Imagens, é um exemplo de atributo nominal, cujos valores são as diferentes classificações de um objeto, tais como “pássaro”, “pato” ou “sapo”. Como determinadas implementações de algoritmos trabalham apenas com atributos nominais, a biblioteca WEKA também implementa procedimentos que permitem discretizar valores numéricos, eliminando esse tipo de limitação.

Quando o domínio de imagens é levado em consideração, campos numéricos, data e *string* não são suficientes para representar todos os tipos de características de imagens. Uma característica, nesse caso, pode ser um vetor de valores, tais como os coeficientes da transformada de *Fourier*, uma fórmula que representa a superfície de uma região do objeto ou ainda uma lista de regiões com sua respectiva relação espacial. Duas abordagens são propostas para contornar esse tipo de problema: criação de subtipos da classe *Attribute* para representar tipos de dados mais complexos, como por exemplo, criando uma classe chamada *2DStringAttribute* para representar uma *2DString*, e; criação de uma classe que faça o mapeamento de um atributo mais complexo para a representação aceita pela biblioteca de mineração. Nos dois casos, é necessária a adaptação do algoritmo de mineração para que o mesmo possa processar o novo tipo de dado, porém, a criação de uma classe de mapeamento diminui esse problema e torna a implementação mais transparente e independente da biblioteca de programação, como será visto na Seção 8.2.

A principal contribuição da biblioteca de mineração, contudo, está nos diversos algoritmos implementados. Apesar de possuir algoritmos de descoberta de regras de associação, o foco desse trabalho está nos algoritmos de classificação e agrupamento. No que se refere a algoritmos de classificação, os mesmos podem ser iterativos (*IterativeClassifier*) ou atualizáveis (*UpdateableClassifier*). No primeiro tipo, a complexidade do modelo gerado aumenta com o tempo ou número de iterações. O segundo tipo é particularmente útil em questões de performance e aplicabilidade em

Sistemas de Recuperação de Imagens por Conteúdo, pois permite que o sistema aprenda incrementalmente, uma instância por vez.

A biblioteca WEKA provê algoritmos de diversas famílias, tais como Bayes (Redes Bayesianas e NaïveBayes), redes neurais, *Instance-based learning* (IBk e *KStar* [Cleary & Trigg 95]), regras de decisão (tabelas de decisão, PART [Witten & Frank 98]) e árvores de decisão (ID3, C4.5). Para permitir que o sistema aprenda dinamicamente, também serão usados algoritmos de agrupamento disponibilizados pela biblioteca WEKA, tais como EM [Dempster et al. 77], COBWEB e *K-means*.

Tanto no caso de classificadores quanto de clusterizadores, o objetivo é recuperar a distribuição ou qualquer outro valor que permita atribuir um grau de pertinência ou peso que uma imagem (instância) tem em relação a uma classe. A princípio, qualquer heurística utilizada por um algoritmo pode ser usada, como uma distribuição de probabilidade ou um valor de entropia. Por esse motivo, os experimentos fazem uso de algoritmos que já implementam alguma função que permite avaliar uma instância, no caso, algoritmos que implementem a interface *DistributionClassifier* e *DistributionClusterer*.

8.1.2 Módulo de Mineração de Imagens

A Figura 9 exibe as principais classes que formam a biblioteca de Mineração de Imagens, que logicamente pode ser dividida em três grupos relacionados entre si, isto é, possuindo classes e componentes comuns. O primeiro grupo refere-se à *base de imagens*, isto é, ao mapeamento das imagens, características e conceitos. O segundo grupo corresponde aos *algoritmos de mineração* e o terceiro representa o *conhecimento* gerado, obtido por meio da mineração.

O objetivo das classes relativas à *base de imagens*, é a ampliação das funcionalidades da biblioteca de Mineração de Dados, enfocando a representação de características de imagens. Cada imagem no banco de dados corresponde a um objeto da classe *ImageInstance*, composto por atributos (*Attributes* e *Features*) e conceitos (*Concepts*). Quem armazena a referência para as imagens é a classe *ImageSet*, responsável por obtê-las do banco de dados e construí-las em memória, seguindo a parametrização dos conceitos e das características (*Features*).

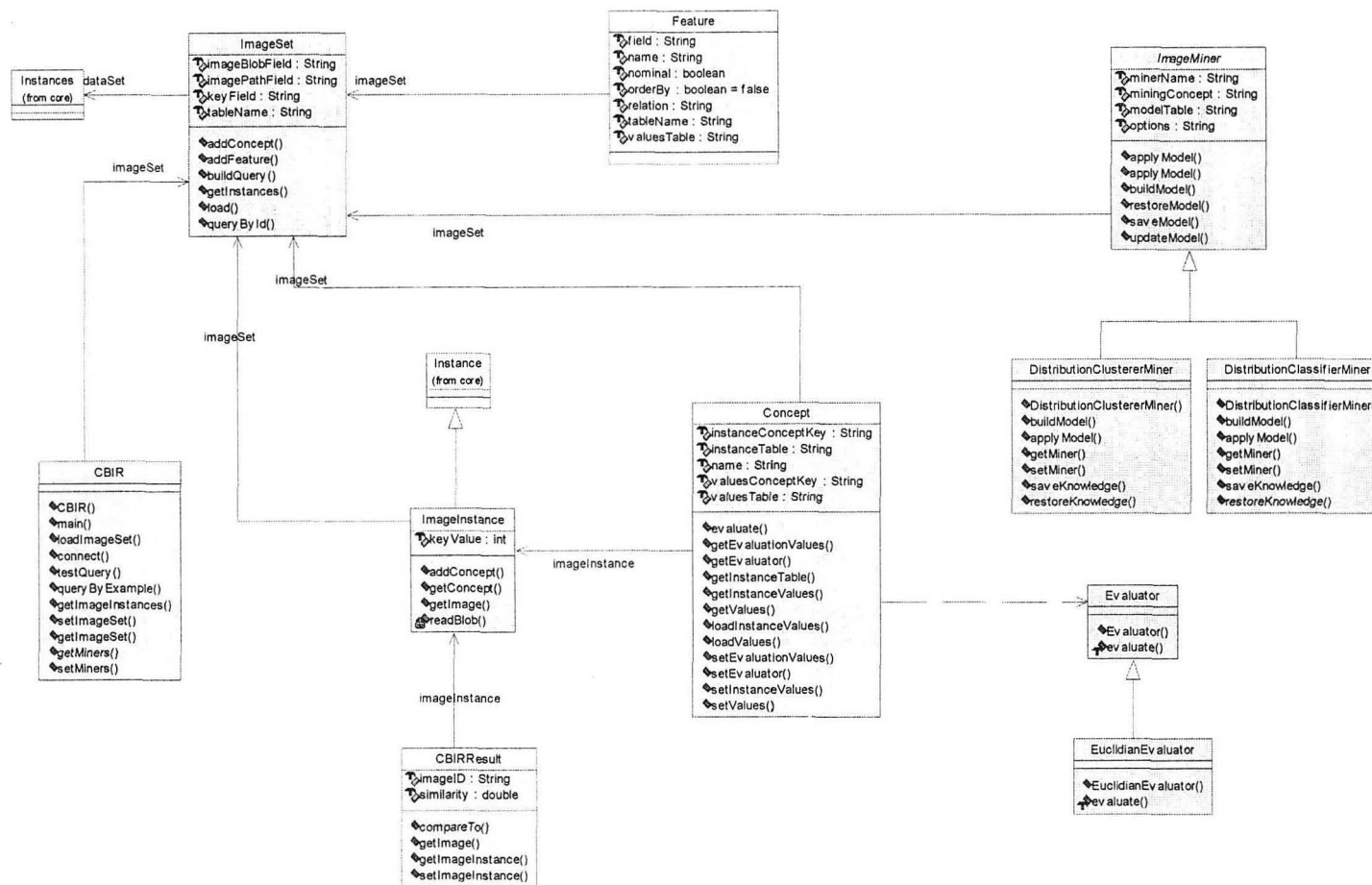


FIGURA 9 – PRINCIPAIS CLASSES, ATRIBUTOS E MÉTODOS DO MÓDULO DE MINERAÇÃO DE IMAGENS E SRIC

Os *algoritmos de mineração* atuam como interface entre o módulo de Mineração de Imagens e o de Mineração de Dados. Dentro desse grupo, encontra-se a classe *ImageMiner*, que possui os métodos necessários para construir o modelo de mineração (uma árvore de decisão, por exemplo) a partir do *ImageSet* e para aplicar o resultado da mineração em cada *ImageInstance*, isto é, atribuir uma avaliação a cada conceito associado. Nesse trabalho, dois tipos de aprendizado são utilizados para a mineração das imagens: supervisionado (*DistributionClassifierMiner*), isto é, depende do rotulamento ou atribuição de conceitos a cada imagem e a posterior aplicação de algoritmos de classificação, e; não-supervisionado (*DistributionClustererMiner*), no qual as classes são geradas pelo próprio sistema através de algoritmos de agrupamento.

Para os algoritmos de mineração, é importante identificar algum atributo que representa o conceito ou a classificação de uma instância. Para a biblioteca WEKA, essa classificação é um atributo (*Attribute*) configurado através da chamada a um método da classe *Instances*. Entretanto, dentro do contexto de dados visuais, foi observada a necessidade de uma implementação que possa trazer uma flexibilidade ainda maior para o sistema, pois uma imagem poderá ter múltiplos conceitos e, para cada conceito, múltiplas atribuições. Com esse intuito, foi modelada a classe *Concept*, parte do terceiro grupo, cujo entendimento deve ficar melhor ilustrado na tabela a seguir:

Nome da Imagem	Área	Total de Regiões	Tipo de Objeto	Tipo de cenário
duck_01	10	5	Pato Pássaro	Paisagem
frog_01	12	6	Sapo	Paisagem Floresta

TABELA 1: EXEMPLO DE CONCEITOS POR IMAGEM

Neste exemplo, cada imagem possui dois conceitos: *tipo de objeto*, que indica qual a natureza do objeto que a imagem representa e *tipo de cenário*, que é o ambiente onde o objeto está. Nada impede que uma imagem possua mais de um objeto ou o mesmo objeto pertença a várias categorias. A mineração permitirá ao

sistema aprender os conceitos de *cenário* e *objeto* para cada imagem, o que será utilizado posteriormente para consultas.

A avaliação atribuída a cada conceito depende da natureza de algoritmo de mineração de dados, como por exemplo, a distribuição de probabilidade de cada imagem pertencer a uma determinada classificação. Isso significa que o resultado da mineração pode ser um vetor de características, um número e até uma *string*. A classe *Concept* é então complementada por uma classe (*Evaluator*) responsável por comparar dois conceitos, o que será imprescindível para o módulo de SRIC.

8.1.3 SRIC

O módulo de Mineração é independente dos demais, ou seja, pode ser utilizado tanto junto ao módulo de SRIC, aqui descrito, quanto junto a um sistema de diagnóstico médico, por exemplo. O componente de recuperação de imagens por conteúdo funciona como ponte entre a camada de apresentação ao usuário e o módulo de Mineração de Imagens, além de executar a recuperação de imagens propriamente dita.

Para executar a tarefa de pesquisa de imagens, esse módulo faz uso de duas classes: *CBIR* e *CBIRResult* (vide Figura 9), além da interface com o usuário que é detalhada na Seção 8.4. A classe *CBIR* cuida da criação de um *ImageSet* contendo as imagens armazenadas no banco de dados. Também guarda referência a todos os *ImageMiners* que serão utilizados para aprender os conceitos configurados para o *ImageSet*. Finalmente, possui uma função que calcula a similaridade entre a imagem-exemplo e cada imagem no banco de dados, utilizando como apoio a classe *Evaluator*, citada no tópico anterior. O resultado é uma lista de objetos da classe *CBIRResult*, que encapsula o *ranking* de similaridade e uma referência à respectiva imagem do banco.

8.1.4 Módulo de Extração de Características

Esse é o módulo que executa a transformação da imagem física (conforme Seção 7) em imagem lógica, isto é, refere-se a extração de características e a sua posterior integração no banco de dados. Inicialmente, a imagem de profundidade é convertida para o formato KDF para que os programas (*glyphs*) desenvolvidos em C utilizando a biblioteca do KHOROS sejam executados, gerando vários arquivos de saída, cada um com diferentes tipos de características. O passo seguinte é executar uma rotina em *Perl* que consolida e interpreta os diversos arquivos e insere (ou atualiza) as imagens e as características no banco de dados. A própria imagem física é armazenada como um campo binário na tabela de imagens.

Esse módulo ainda não está integrado ao sistema de mineração, sendo executado através de linha de comando (Unix *shell* e *Perl*), porém já está configurado para importar todas as vistas de cada objeto em uma única execução.

8.1.5 Apresentação e interação

Por ser uma solução flexível, o aproveitamento da biblioteca de Mineração de Imagens não está presa à Internet ou a programas *desktop* ou *client/server*. No caso específico do SRIC3D, é importante que o acesso ao aplicativo seja via Web. Entretanto, para determinados graus de interatividade, a interface HTML é bastante limitada. Foge do escopo desse trabalho determinar qual a melhor interface de usuário e de que modo disponibilizá-la.

8.2 DETALHAMENTO

As principais classes do módulo de Mineração de Imagens serão descritas nessa Seção, a fim de se obter uma compreensão da solução proposta e da implementação do sistema.

8.2.1 *ImageSet*

Armazena uma referência para o objeto *Instances*, que por sua vez, aponta para cada um dos *ImageInstances* ou imagens. A principal funcionalidade dessa classe é converter as imagens do banco de dados em instâncias, de acordo com a configuração das *Features* e dos *Concepts*. As principais propriedades são:

- a) *tableName* e *keyField*: identificam a tabela e a chave primária da tabela em que estão armazenadas as imagens, sendo usado posteriormente para a montagem da consulta SQL que recupera e atualiza os dados do banco;
- b) *features*: lista de características de imagens, que serão convertidas em *Attributes* – cada característica possui informações que também irão auxiliar a montagem das instâncias;
- c) *concepts* : armazena a configuração de cada conceito associado a uma imagem, para depois permitir a consulta ao banco de dados.

Os principais métodos são:

- a) *buildQuery*: utiliza a parametrização das características e dos conceitos para gerar a consulta SQL utilizada pelo método *getInstances*;
- b) *getInstances*: efetivamente cria os objetos da classe *ImageInstance*.

A atual implementação trabalha com todas as instâncias em memória, isto é, para que os algoritmos sejam executados é preciso carregar todo o conteúdo do banco de dados, o que para grandes aplicações é inviável. A solução é carregar as instâncias sob demanda ou apenas manter em memória uma referência (a chave primária por exemplo) para o registro da tabela. Para isso, é necessário adaptar a implementação da classe *Instances* e *ImageInstance*. O trecho de código abaixo exemplifica como é a utilização da classe *ImageSet*, demonstrando a configuração da tabela de imagens e sua chave primária, além da conexão ao banco de dados:

```
imageSet = new ImageSet(connection);  
imageSet.setTableName("v_image");
```

```
imageSet.setKeyField("id_image");
```

8.2.2 *ImageInstance*

Especialização de uma *Instance*, cada *ImageInstance* representa uma imagem do banco de dados e possui um ou mais conceitos associados, que posteriormente serão atualizados pelo módulo de mineração. Os principais métodos permitem a atualização dos atributos e dos conceitos, como segue:

- a) *getValue* e *setValue* – recupera ou atualiza os valores de cada atributo associado a instância;
- b) *getKeyValue* e *setKeyValue* – recupera ou atualiza o valor do identificador da imagem no banco de dados;
- c) *getImage* – obtém a representação física da imagem.

8.2.3 *Feature*

Uma característica pode assumir vários valores, tais como: um número; uma descrição; um campo associado a outra tabela; pode ser o produto de um processamento de uma imagem; pode ser o resultado de uma consulta em outras tabelas, ou; um conceito gerado por um outro processo de mineração. Para dar maior flexibilidade ao sistema, a classe *Feature* abstrai uma característica de imagem, permitindo mapeá-la em um ou mais atributos ou ainda guiar o processamento de algoritmos de mineração (módulo de Extração de Características). A fim de executar essa tarefa, algumas propriedades devem ser configuradas:

- a) *field*: *campos* do banco de dados em que a característica é mapeada, permitindo que uma *Feature* possa ser armazenada em um ou mais campos ou tabelas;
- b) *values*: uma feature pode possuir mais de um valor possível – caso de atributos nominais;
- c) *tableName*: para auxiliar a construção da consulta SQL.

Na arquitetura proposta, apenas uma implementação simples de *Feature* é disponibilizada, fazendo um mapeamento direto para a classe *Attribute*. Entretanto, implementações mais complexas podem incluir outras propriedades que podem ser configuradas para alterar o comportamento dos algoritmos de mineração. Uma classe interessante poderia ser a *FourierDescriptor*, que recuperaria os descritores obtidos pela transformada de *Fourier* ou ainda uma característica chamada *Region*, que armazenaria todas as regiões que compõem uma imagem juntamente com sua distribuição espacial. De modo a tornar mais claro, o trecho de código abaixo demonstra como associar uma característica (no caso, o segundo momento angular) a um *ImageSet*:

```
imageSet.addFeature(new Feature("ang_sec_moment", connection));
```

8.2.4 Concept

Cada objeto *Concept* permite quantificar e qualificar cada classe associada a uma instância, bem como, armazenar o resultado da mineração e comparar o mesmo conceito em relação a diferentes imagens. Antes, é preciso configurar as seguintes propriedades:

- a) *instanceTable* e *instanceConceptKey*: nome da tabela e do campo que armazenam os valores de conceito por instância;
- b) *valuesTable* e *valuesConceptKey*: nome da tabela e do campo que possuem os valores possíveis de um conceito, semelhante a um atributo nominal;
- c) *instanceValues*: os valores atribuídos a uma determinada instância;
- d) *values*: os valores possíveis do conceito, obtido a partir de uma consulta a tabela apontada por *valuesTable*;
- e) *evaluationValues*: armazena a distribuição de probabilidade ou qualquer outro valor que indique um grau de avaliação do conceito.

Os métodos principais e suas funcionalidades são descritos abaixo:

- a) *setEvaluator* e *getEvaluator*: registra um *Evaluator*, que é uma classe que implementa um algoritmo de avaliação de um conceito e, no caso, um conceito pode ter vários “avaliadores”, normalmente um para cada *ImageMiner*;
- b) *loadValues* e *loadInstanceValues*: recupera os valores possíveis do conceito e os valores por instância, a partir do banco de dados;
- c) *evaluate*: compara o mesmo conceito entre duas imagens, para cada um dos avaliadores registrados, geralmente calculando a distância (similaridade).

O trecho de código a seguir configura um conceito:

```
Concept concept = new Concept(connection);
concept.setName("object_type");
concept.setInstanceTable("image_object_type");
concept.setValuesTable("object_type");
concept.setValuesConceptKey("id_concept");
concept.setInstanceConceptKey("id_concept");
```

8.2.5 Evaluator

Permite comparar dois objetos quaisquer. Isso significa que um objeto dessa classe pode ser implementado de modo a comparar duas *Features*, duas imagens ou dois conceitos, podendo ser usado por qualquer outra classe, principalmente a classe *Concept* e a classe *CBIR*. Para isso, basta implementar o método *evaluate*.

Já é disponibilizado como exemplo a classe *EuclidianEvaluator*, que calcula a distância geométrica entre duas avaliações de conceito (*evaluationValues*), assumindo que os objetos comparados são vetores que possuem a distribuição de probabilidade de uma instância em relação a cada classe do sistema.

O código a seguir mostra como associar um *Evaluator* a um conceito:

```
concept.setEvaluator("KStar", new EuclidianEvaluator());
```

```
concept.setEvaluator("IBk", new EuclidianEvaluator());  
concept.setEvaluator("EM", new EuclidianEvaluator());
```

8.2.6 *ImageMiner*, *DistributionClassifierMiner* e *DistributionClustererMiner*

A classe *ImageMiner* tem como responsabilidade interagir com o algoritmo de mineração, guiando-o no processo de extração de conhecimento de imagens. As principais responsabilidades dessa classe são:

- a) *buildModel*: gera um modelo – regras, instâncias, árvores de decisão, redes neurais ou de Bayes – a partir de um banco de imagens (*ImageSet*);
- b) *saveModel*, *retrieveModel*, *updateModel*: armazena o modelo no banco, recupera-o e atualiza-o, de modo que o sistema evolua conforme a variação do número de imagens;
- c) *applyModel*: aplica o modelo em cada uma das imagens do banco e em imagens novas, atribuindo uma avaliação aos conceitos em que o minerador estiver associado.

A biblioteca de mineração já traz duas implementações genéricas: *DistributionClassifierMiner* e *DistributionClustererMiner*. A primeira, interage com qualquer algoritmo da biblioteca WEKA que implementa a interface *DistributionClassifierMiner* (como *KStar* ou *IBk*), isto é, que permite atribuir uma distribuição de probabilidade a cada conceito. A segunda, faz o mesmo tipo interação, porém com algoritmos que implementam a interface *DistributionClustererMiner*, como é o caso EM (*Expectation Maximization*).

O modo de utilização do *ImageMiner* pode ser visto no trecho de código a seguir:

```
IBk ibkClassifier = new IBk();  
ibk=new DistributionClassifierMiner(connection,"IBk",ibkClassifier);
```

Essa outra parte do código mostra como gerar e aplicar o modelo a todas instâncias da base de imagens e também a uma instância específica:

```
ibk.buildModel();  
ibk.applyModel();  
ImageInstance img = (ImageInstance) imageSet.queryById(imageID);  
em.applyModel(img);
```

A principal contribuição do minerador é a possibilidade de aplicar o modelo a uma imagem qualquer. Isso significa que uma classe que implementa os métodos *buildModel*, *applyModel*, *saveModel*, *retrieveModel* e *updateModel* pode ser integrada facilmente ao módulo de Mineração de Imagens. Poderia ter sido criada uma classe *IBkImageMiner*, por exemplo, que fosse capaz de interpretar características como *2DString*, sem a obrigatoriedade de utilizar a implementação feita pela biblioteca WEKA.

8.2.7 CBIR e CBIRResult

Essas duas classes atuam em conjunto e correspondem à implementação da lógica que envolve um SRIC apoiado sobre um módulo de Mineração de Imagens. A principal propriedade da classe *CBIR* é a configuração de múltiplos mineradores. Isso significa que um mesmo conceito poderá ser avaliado por diferentes implementações de *ImageMiner* e ao final, cada uma irá contribuir para indicar o quanto cada conceito pertence a uma determinada classificação. A classe *CBIR* é então configurada com o *ImageSet*, que por sua vez é composto de *Features* e *Concepts*. A cada combinação de minerador e conceito, deve ser indicado um objeto do tipo *Evaluator*. Uma vez parametrizado todos esses itens, o sistema permitirá dois tipos de consulta:

- a) *QueryByExample*: recebe uma instância de uma imagem, recuperada diretamente do banco de dados ou obtida por meio do módulo de extração de características, que será usada como exemplo ou base de comparação;

- b) *QueryByFeatures*: recebe os valores das características e internamente gera objetos da classe *ImageInstance* e convenientemente aplica o método *QueryByExample*.

Os dois métodos acima retornam as imagens ordenadas segundo um *ranking* de similaridade, que é a própria classe *CBIRResult*, responsável por interfacear com uma *ImageInstance*.

O diferencial de um sistema SRIC que utiliza Mineração de Imagens está no modo como a consulta é executada. Ao invés de utilizar uma função de similaridade baseada em uma heurística pré-estabelecida, o SRIC proposto aprende essa função, com as seguintes vantagens:

- a) o sistema pode evoluir, isto é, o modelo refina o conhecimento gerado conforme novas instâncias são inseridas;
- b) o sistema é genérico, aprendendo de maneira independente das características, ou seja, se for incluída uma nova característica, ela poderá influenciar ou não no comportamento do modelo sem necessidade de reprogramação;
- c) a pesquisa é feita tomando por base conceitos, isto é, de uma maneira mais próxima ao raciocínio humano;
- d) o modelo aprendido pelos algoritmos de mineração podem ser aplicados a uma imagem já cadastrada no banco ou a uma imagem recém processada pelo módulo de extração de características;
- e) o sistema pode aprender com erros, isto é, através de mecanismos de realimentação de relevância, uma imagem pode ser re-classificada, ensinando ao sistema um novo conceito;
- f) o sistema pode criar uma nova classificação das imagens, através dos algoritmos de aprendizado não-supervisionado.

8.3 MODELO DE IMPLEMENTAÇÃO

O objetivo dessa Seção é apresentar como todos os componentes apresentados na Seção 8.2 interagem. O primeiro item detalha o funcionamento dos módulos de Mineração e de Consulta de imagens sob o ponto de vista do relacionamento dos seus componentes, numa visão seqüencial e estática. O segundo item destaca como deve ser a implementação da arquitetura para prover um sistema dinâmico e completo de consulta e mineração, já evidenciando alguns desafios a serem enfrentados.

8.3.1 Visão estática

Para tornar mais clara a explanação, será utilizada como referência a Figura 10, que é um diagrama de seqüência que detalha as chamadas dos principais métodos, passagem de parâmetros e instanciação de objetos.

O bloco “*User Interface*” representa a interface com o usuário, que pode ser uma interface gráfica (vide Seção 8.4) ou o código fonte, caso o usuário esteja utilizando a biblioteca de Mineração de Imagens diretamente. É a interface com o usuário que será responsável por instanciar os principais componentes do sistema.

O primeiro passo consiste em criar uma base de imagens (*ImageSet*) e configurá-la. Nesse caso, foi configurada a tabela e a chave primária. Em seguida, cada uma das características deve ser criada separadamente, para isso parametrizando-se o nome da *Feature* e o a campo e tabela no banco de dados em que cada característica está armazenada (no diagrama, apenas uma *Feature* foi gerada). De maneira semelhante, configura-se cada um dos conceitos (*Concept*), indicando as tabelas para recuperação dos valores possíveis, bem como o componente de avaliação.

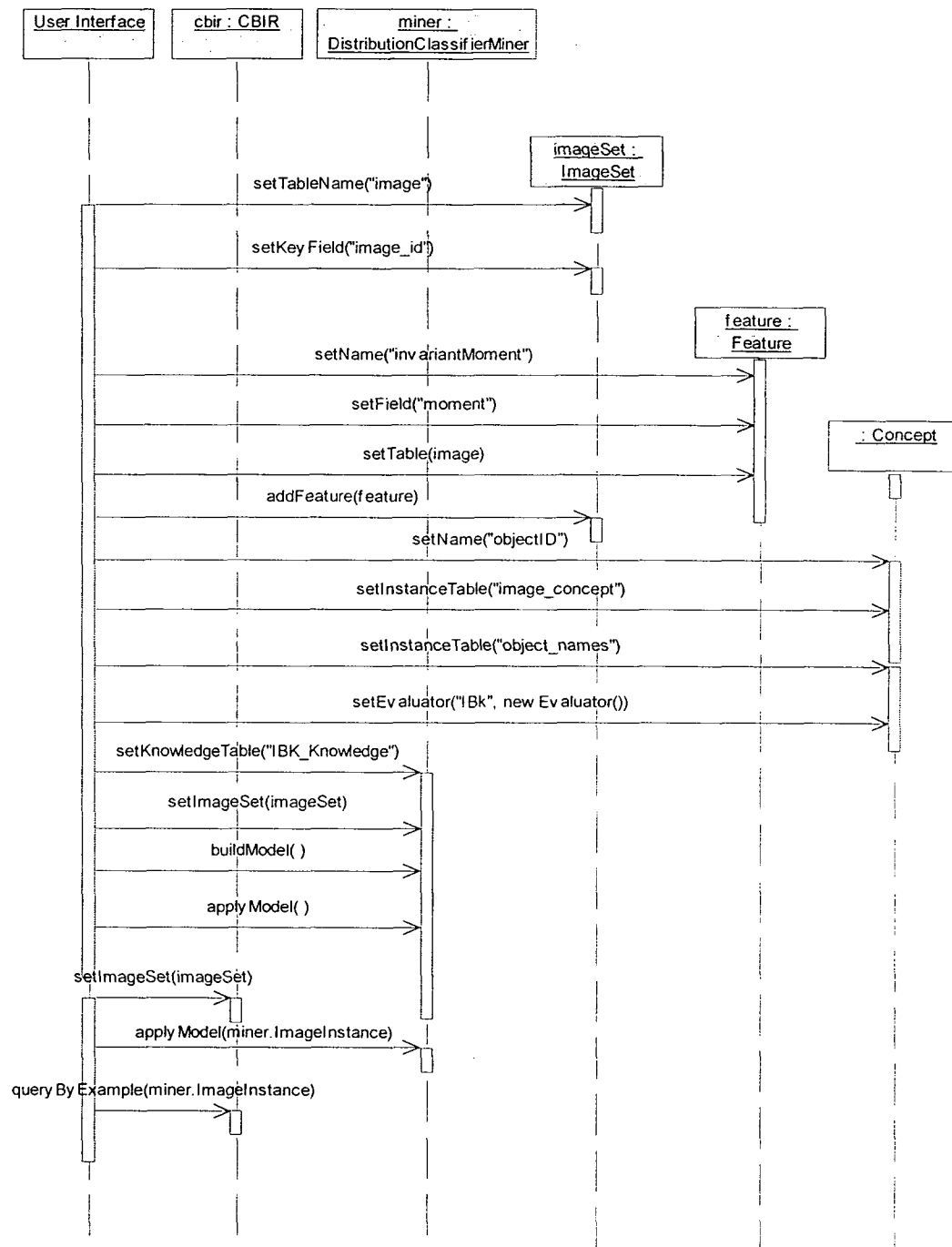


FIGURA 10 – DIAGRAMA DE SEQÜÊNCIA DO MÓDULO DE CBIR

Na seqüência, devem ser instanciados os algoritmos de mineração (*ImageMiner*) e registrados junto ao componente de consulta, o *CBIR*. Antes de executar a pesquisa, o modelo já deve estar construído e aplicado a cada imagem do banco de dados. A consulta então procede, retornando um vetor de *CBIRResult*.

8.3.2 Visão Dinâmica

No tópico anterior, foi apresentado como os componentes interagem, desde o instante em que um *ImageSet* é criado até o momento em que uma consulta é executada. Entretanto, essa abordagem ignora o fato de que a arquitetura proposta prevê que o sistema deve evoluir conforme novas instâncias e características são inseridas ou atualizadas, isto é, não possui um comportamento necessariamente seqüencial. Da mesma forma que um banco relacional possui *triggers* que são ativadas toda vez que um registro de uma tabela é modificada, propagando o evento para outras partes do banco, uma estratégia semelhante deve ser adotada nesse caso.

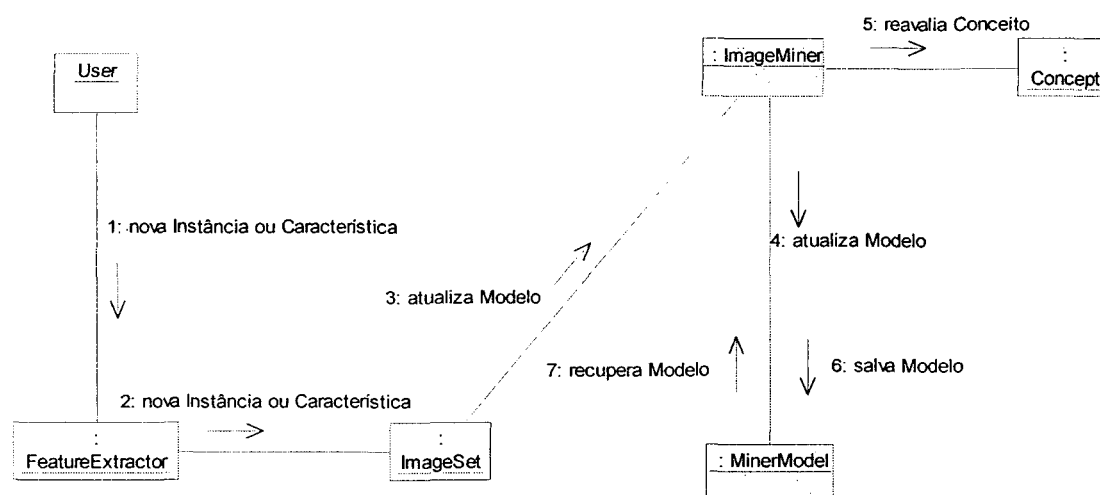


FIGURA 11 – DIAGRAMA DE COLABORAÇÃO DO MÓDULO SRIC

Na Figura 11, no momento em que uma nova característica ou instância é criada, é necessário atualizar o componente de mineração, para que o mesmo possa

aprender com essa nova instância, possivelmente tendo que reaplicar o conhecimento em todos os conceitos e instâncias já avaliados. O que a princípio era uma atividade simples, acaba gerando impacto em todo o sistema.

Quando uma instância é criada e armazenada, o sistema precisa avaliar se o modelo deve ser atualizado ou não. Se a instância representa um novo conceito, apenas o modelo para aquele conceito deve ser criado; se a instância está relacionada com um conceito já aprendido, o sistema deve avaliar se a nova instância já pode ser classificada com o modelo atual. Se já pode ser classificada, não há necessidade de refazer o modelo, embora esse novo item pode ser útil para refinar o modelo e melhorar o entendimento do conceito. Se houve erro na classificação, é possível que o modelo deva ser recriado, o que é um processo oneroso, pois envolve reprocessar (*buildModel*) todas as instâncias no banco de dados (já filtrando por conceito) e reaplicar o conhecimento em cada uma (*applyModel*), a fim de obter a distribuição de probabilidade. Isso significa reaplicar um ou mais algoritmos de mineração, considerando dezenas de características e milhares de imagens.

Um desafio que surge desse cenário é a melhoria da performance de algoritmos para tratar da atualização do modelo – alguns algoritmos, por exemplo, permitem uma atualização iterativa, outros não. Outro aspecto importante é o armazenamento do modelo. Se a base de dados for muito grande, não é recomendado refazê-lo toda vez que o sistema de mineração de imagens é inicializado. A flexibilidade do sistema deve estar, portanto, na capacidade de salvar e recuperar o modelo, ou seja, persistir o conhecimento no banco de dados.

Finalmente, um outro aspecto dinâmico que a implementação da arquitetura deve prever, é a performance na consulta e indexação. Uma vez que a pesquisa será feita sobre o conhecimento gerado, é importante o uso de uma estrutura de índice que comporte os vetores de características (pesos, distribuição de probabilidade) obtidos pela aplicação da mineração de imagens em cada conceito. Além disso, esse índice deve ser dinâmico o suficiente para permitir a sua atualização toda vez que o modelo em memória é modificado.

8.4 IMPLEMENTAÇÃO E EXPERIMENTOS

Com o intuito de demonstrar uma implementação da arquitetura proposta, foi desenvolvido um aplicativo em Java com Swing para facilitar os testes de mineração e servir de protótipo funcional (isto é, evolutivo) para o subsistema de Mineração de Imagens do SRIC3D. A ênfase dessa versão é exercitar o comportamento dos algoritmos de mineração e ratificar a usabilidade do método, sendo composta de três partes, cada uma representada por um “*tab*” ou “*tabsheet*” do programa (vide Figura 12): *Database*, que configura as características e conceitos de imagens; *Miner*, que parametriza os diferentes mineradores, e; *Query*, que permite a execução de consultas.

8.4.1 Configuração da base de dados

Conforme a Figura 12, esse painel ou “*tab*” é composto de três subdivisões. Na primeira divisão, *Image Set Properties*, o sistema permite configurar em que tabela estão as imagens na base de dados (*Table Name* e *Key Field*), bem como, o campo utilizado para obter a imagem física (*Blob Path Field* e *Blob Field*). As imagens estão armazenadas num base de dados PostgreSQL, conforme o esquema da Figura 13.

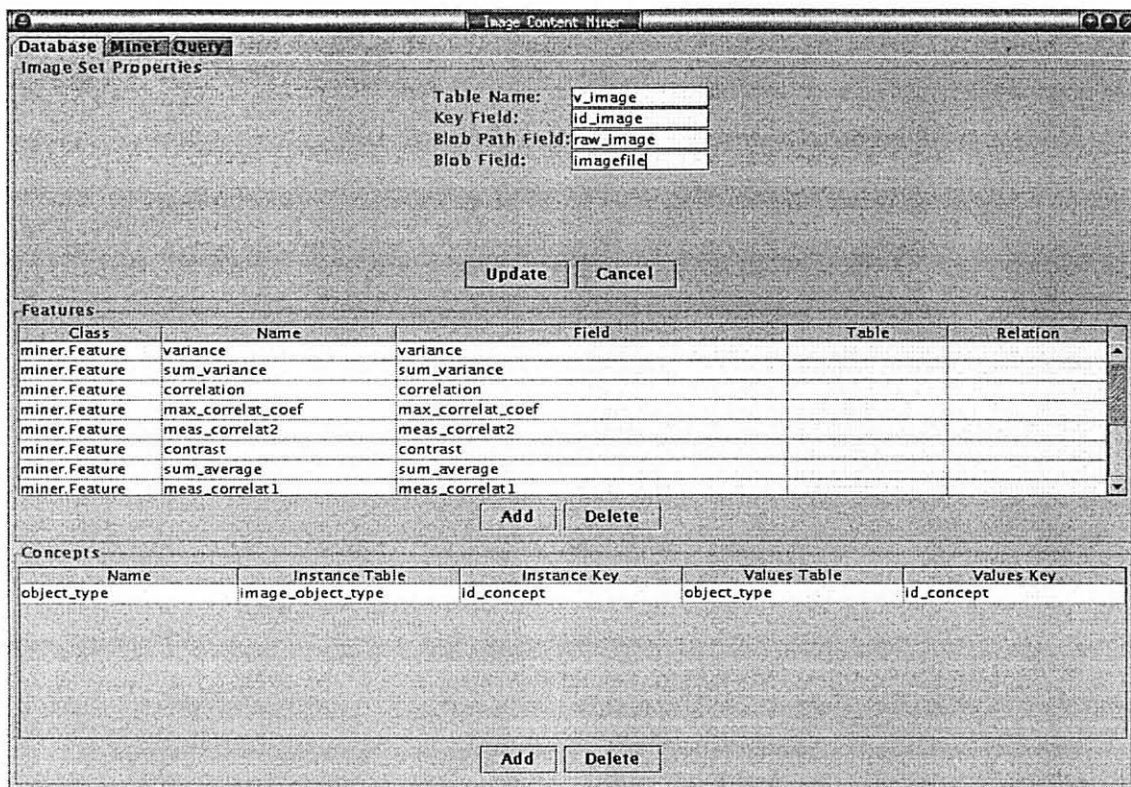


FIGURA 12 – CONFIGURAÇÃO DO BANCO DE DADOS

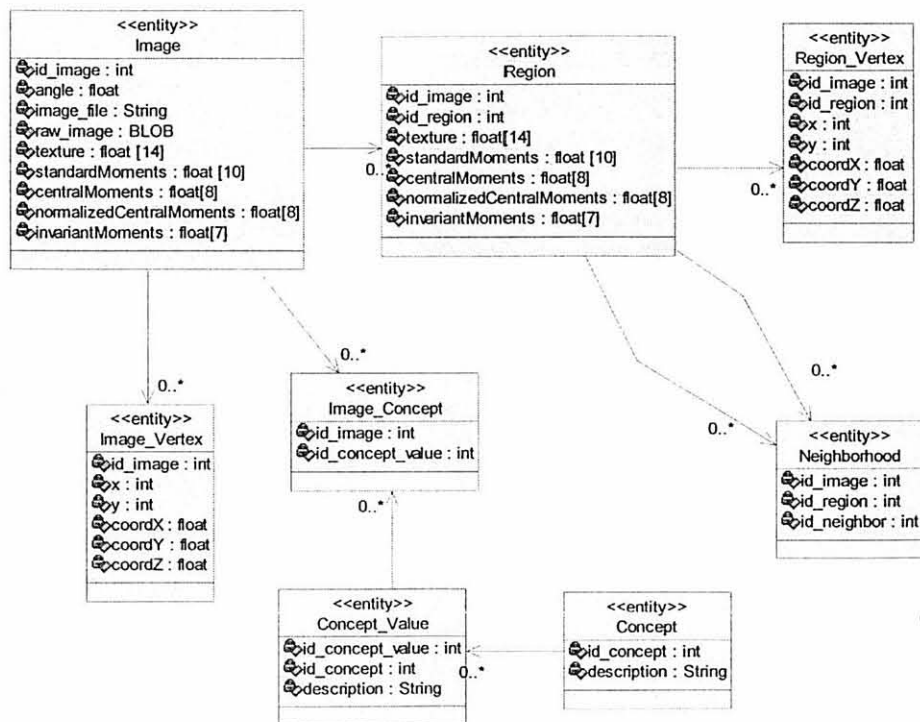


FIGURA 13 – MODELO DE DADOS

A Figura 13 exibe uma visão lógica, isto é, uma representação em alto nível do modelo Entidade-Relacionamento das tabelas do banco de dados PostgreSQL. O objetivo do diagrama é demonstrar as associações entre tabelas (identificadas pelo estereótipo <<entity>>) e os principais campos.

Cada imagem (tabela *Image*) possui um identificador único, além das características (tais como textura e momentos), ângulo da vista do objeto e a imagem física. O contorno da imagem é armazenado na tabela *Image_Vertex* através dos pontos que compõem o polígono que forma o contorno (tanto as coordenadas reais quanto em pixels). Oriundo da segmentação da imagem, cada região (tabela *Region*) também é identificada unicamente e as mesmas características de textura, momento e contorno (tabela *Region_Vertex*) também são armazenadas. Para facilitar futuros testes, a informação de vizinhança de regiões é obtida e guardada na tabela *Neighborhood*. Se novas características forem geradas, novos atributos deverão ser criados de maneira consistente nas tabelas de região e de imagem.

Diretamente relacionada com o módulo de mineração e SRIC, as tabelas *Image_Concept*, *Concept_Values* e *Concept* representam as possíveis classes ou conceitos que cada imagem no banco pode assumir. A primeira tabela armazena os conceitos por imagem, a segunda tabela armazena os valores disponíveis por conceito e a terceira tabela, os tipos de conceitos (nesses experimento, foi trabalhado apenas o conceito tipo de objeto ou *object_type*).

Ainda na Figura 12, a segunda subdivisão permite ao usuário editar, adicionar e excluir *Features*. Para cada característica, o usuário também deve informar a classe que implementa a *Feature*, isto é, deve ser informada uma classe que herda de *miner.Feature* e encontra-se no CLASSPATH¹², bem como o nome da característica e informações sobre sua localização no banco de dados. Isso traz a flexibilidade de incluir *Features* mais complexas e parametrizar cada uma, incluindo pesos ou critérios que possam influenciar os algoritmos de mineração. Nesse

¹² CLASSPATH: variável de ambiente que indica os diretórios (semelhante ao PATH do DOS e UNIX) nos quais a linguagem Java irá procurar qualquer classe que possa ser carregada dinamicamente

experimento, foram utilizadas 14 características obtidas a partir do vetor de co-ocorrência.

A terceira subdivisão é utilizada para indicar que conceitos cada imagem possui, isto é, em que tabela e campo estão armazenados os conceitos por *ImageInstance*. Nesse exemplo, foi configurado o conceito *object_type*, que é o nome ou o tipo do objeto associado a cada imagem. A lista de valores possíveis está especificada na tabela 2.


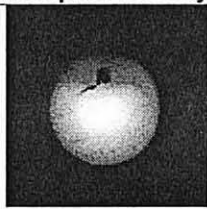
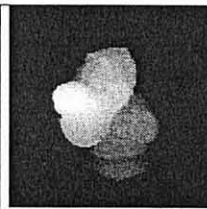
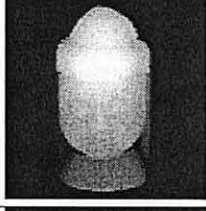

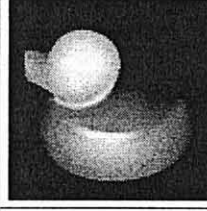
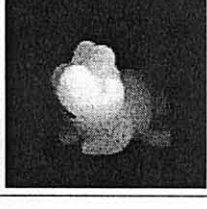
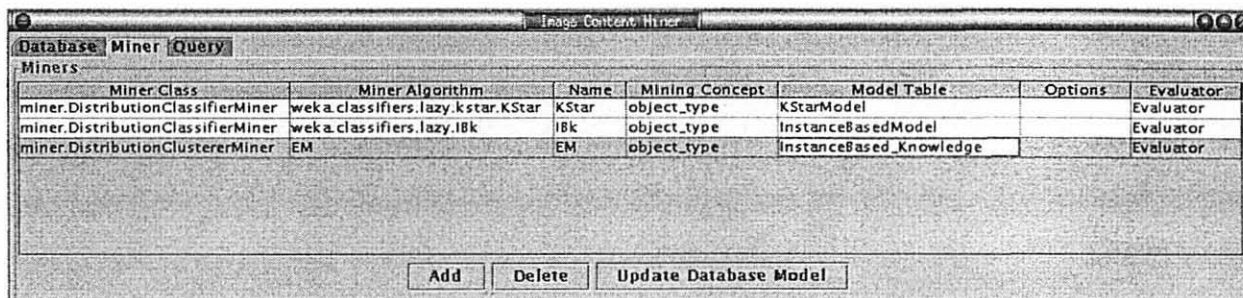
Exemplos de Objetos e Conceitos		
	Anjo	
	Maçã	
	Pássaro	
	Buda	
	Lagosta	
	Pato	
	Sapo	

TABELA 2: EXEMPLOS DE OBJETOS ARMAZENADOS NO BANCO DE IMAGENS

8.4.2 Configuração dos algoritmos de mineração

Nessa parte do aplicativo, o usuário pode configurar cada um dos mineradores (vide Figura 14). Nesse caso, três mineradores foram utilizados: IBk e *KStar*, que são algoritmos de aprendizado supervisionado do tipo *Instance-based learning* e um clusterizador, o EM. Foram parametrizados dois tipos de *ImageMiner*, um para classificadores e o outro para *clusterers* (clusterizadores), mas nada impede que sejam utilizados algoritmos customizados: basta informar qualquer classe que

implemente os métodos previstos na classe pai *ImageMiner* e incluir o arquivo no CLASSPATH.



The screenshot shows a window titled "Image Content Miner" with a "Database Miner Query" tab. It displays a table of configured miners. The table has seven columns: Miner Class, Miner Algorithm, Name, Mining Concept, Model Table, Options, and Evaluator. There are three rows of data. Below the table are three buttons: "Add", "Delete", and "Update Database Model".

Miner Class	Miner Algorithm	Name	Mining Concept	Model Table	Options	Evaluator
miner.DistributionClassifierMiner	weka.classifiers.lazy.kstar.KStar	KStar	object_type	KStarModel		Evaluator
miner.DistributionClassifierMiner	weka.classifiers.lazy.IBk	IBk	object_type	InstanceBasedModel		Evaluator
miner.DistributionClustererMiner	EM	EM	object_type	InstanceBased_Knowledge		Evaluator

FIGURA 14– CONFIGURAÇÃO DOS ALGORITMOS DE MINERAÇÃO

Um mesmo *ImageMiner* pode estar presente mais de uma vez nessa relação, desde que seja atribuído um nome diferente. É interessante observar que cada minerador pode estar associado a um conceito diferente (*Concept*). Nesse exemplo, todos os algoritmos estão relacionados com o mesmo *Mining Concept*, *object_type* e o mesmo *Evaluator*, no caso, *EuclidianEvaluator*.

Finalizada a configuração do banco de dados e dos algoritmos, o usuário pode pressionar o botão *Update Database Model*, que irá chamar os métodos *buildModel* e *applyModel* de cada minerador configurado, atualizando o modelo de cada algoritmo e criando a base de conhecimento necessária para execução das consultas.

8.4.3 Execução de consultas

No painel *Query Specification* (Figura 15), o usuário pode executar dois tipos de consulta: a consulta por intervalo (*range query*), em que o usuário pode especificar o conteúdo das características através de uma sintaxe semelhante à cláusula WHERE de uma consulta SQL, ou; a consulta por similaridade, na qual o usuário escolhe uma imagem a partir da tabela *Image Browser*, que lista todas as imagens do banco de dados. Assim que o módulo de extração de características estiver integrado ao subsistema de mineração, o usuário poderá especificar qualquer imagem de profundidade como exemplo, pois a mesma será importada e terá suas características extraídas e avaliadas para servirem de parâmetro de pesquisa.

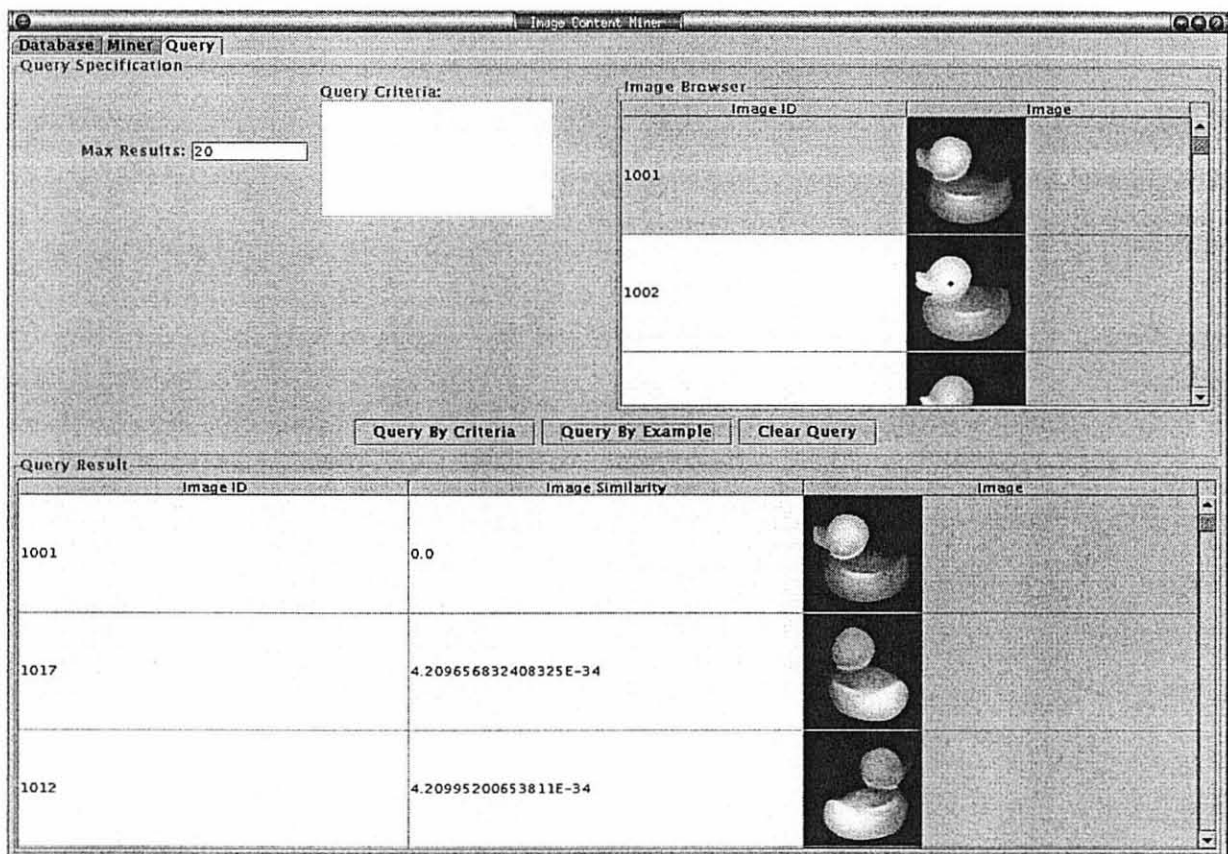


FIGURA 15 – ESPECIFICAÇÃO E VISUALIZAÇÃO DE CONSULTA

O resultado da consulta é exibido logo abaixo, no painel *Query Result*. Para proceder a consulta, o seguinte algoritmo é executado:

- a) aplica-se cada minerador configurado na imagem a ser pesquisada (critério ou exemplo);
- b) o sistema compara a imagem original com cada imagem armazenada no banco, calculando a similaridade entre as mesmas, conforme os passos abaixo:
 - ◆ para cada conceito, é executado o método *evaluate* confrontando a imagem exemplo/critério com a imagem no banco;
 - ◆ para cada minerador associado ao conceito, efetua-se o cálculo da similaridade, comparando-se os respectivos *evaluationValues* (distribuição de probabilidade) de cada algoritmo;
 - ◆ conforme a implementação da classe *CBIR*, o valor de cada similaridade (por minerador) será utilizado para uma média ponderada ou uma simples soma;
 - ◆ os valores obtidos em cada conceito também serão somados ou utilizados para um cálculo de média ponderada;
- c) o sistema ordena as imagens em ordem crescente de similaridade.

No exemplo da Figura 15, naturalmente a imagem utilizada como exemplo possui valor 0 (zero), indicando igualdade. Logo em seguida, aparecem outras faces do mesmo objeto, com uma diferença bem pequena no valor de similaridade. A fim de flexibilizar o sistema e tornar mais prático o seu uso, é possível refinar o mecanismo de consulta de várias maneiras, tais como:

- a) pode ser atribuído um peso a cada algoritmo de mineração, permitindo diminuir ou aumentar a influência de uma determinada avaliação. Por exemplo, pode ser interessante atribuir um peso menor aos algoritmos de agrupamento e um peso maior aos algoritmos de árvore de decisão;
- b) da mesma maneira, o sistema também permite atribuir um peso para cada conceito;
- c) configurações diferentes podem ser feitas tanto na classe *Evaluator* quanto na classe *CBIR*, mudando o comportamento do mecanismo de

consulta sem impactar no conhecimento gerado pelos mineradores, ou seja, esses novos parâmetros têm influência apenas no momento da consulta;

- d) as *Features* também podem ser parametrizadas, interferindo na execução dos algoritmos de mineração, possivelmente aumentando a importância de algumas características em detrimento das demais.

Como já foi evidenciado no decorrer do trabalho, a separação das diferentes responsabilidades do sistema (acesso aos dados, mineração, consulta, similaridade) em várias classes, confere à solução um grande poder de expansão e parametrização. A proposta é que essa praticidade seja alcançada através de especializações das classes modeladas, principalmente a inclusão de novas funcionalidades na classe *Features* (para melhor representar atributos visuais) e na classe *ImageMiner* (para melhor adaptar os algoritmos de mineração para o contexto de imagens).

8.4.4 Resultados experimentais

Com o objetivo de demonstrar a solução desenvolvida através de exemplos práticos, foi criada uma base de testes com 126 imagens distribuídas em 7 conceitos, isto é, 18 imagens por tipo de objeto e 14 características de textura. As características de textura foram obtidas a partir da imagem correspondente ao eixo Z, mas futuramente, será utilizada a imagem das normais, além das características especificadas na Seção 7. Em seguida, foi gerada uma base de conhecimento (modelo) para 5 algoritmos de mineração, fazendo uso de 66% da base de testes com instâncias escolhidas aleatoriamente. A seguir, cada modelo foi aplicado sobre as instâncias do banco, gerando uma distribuição de probabilidade por algoritmo. Esse resultado da mineração serve então de subsídio para a execução de consultas utilizando uma imagem-exemplo, confrontando-a com cada instância na base de imagens. Neste trabalho, será apresentado o resultado da execução de 5 algoritmos individualmente e uma avaliação geral. Todos os testes apresentados aqui utilizaram a imagem 1002 como exemplo.

Deve ficar claro que a intenção destes experimentos é obter uma visão inicial das possibilidades da ferramenta, isto é, de maneira empírica estabelecer estratégias de mineração que podem ser adotadas especialmente no projeto SRIC3D. Essa tarefa envolve analisar o comportamento de alguns algoritmos, verificar a distribuição de probabilidade que cada um é capaz de calcular e avaliar a usabilidade da mineração na recuperação de imagens por conteúdo. Para avaliação de performance e precisão, é necessária uma base de dados maior, evolução que será feita para atender a especificação do SRIC3D.

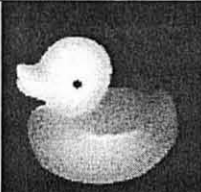



#	ID	Similaridade	Classificação	Imagem
1	1002	0.0	{ 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 }	
2	1012	0.0	{ 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 }	
19	3016	1.274754878398196	{0.0, 0.25, 0.75, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 }	
23	6003	1.414213562373095	{ 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 }	

TABELA 3: MINERAÇÃO DE IMAGENS COM O ALGORITMO PART


A fim de ilustrar cada experimento, é exibida uma tabela com as principais instâncias retornadas pelo mecanismo de busca, identificadas pelos seguintes campos:

- Ranking* (coluna #): a colocação da imagem em relação a imagem-exemplo;
- ID*: identificação da imagem no banco de dados;

- c) *Similaridade*: valor da similaridade, obtido a partir da distância Euclidiana calculada sobre a imagem-exemplo e a imagem no banco, usando a distribuição de probabilidade como parâmetro de comparação;
- d) *Classificação*: vetor de distribuição de probabilidade - para os algoritmos de classificação, o primeiro item corresponde à classe *pato*, o segundo corresponde à classe *sapo*, e assim por diante; para os algoritmos de agrupamento, a classe é inferida pelo próprio algoritmo.

O primeiro teste (vide Tabela 3) foi feito com o algoritmo PART , algoritmo de lista de decisão [Witten & Frank 98] que apresentou um grau de discriminação maior, pois atribui valores mais próximos de 1 (um) para a classe correta da imagem e 0 (zero) para as demais. Para determinados tipos de consulta, isso pode ser um problema, pois a classificação feita pelo modelo pode estar incorreta, trazendo muitas imagens com classes diferentes da original. Na tabela a seguir, todas as imagens do conceito *pato* tiveram 100% de similaridade. Entre as posições 19 e 23, a similaridade foi influenciada porque o algoritmo atribuiu a cada imagem duas classes. Nas demais imagens, a similaridade manteve o mesmo valor.

O segundo teste (Tabela 4), feito com redes neurais, apresenta instâncias com valores de probabilidade mais heterogêneos, o que confere uma classificação mais distribuída entre os conceitos (o que pode ser interessante para determinados tipos de consulta).

#	ID	Similaridade	Classificação	Imagem
1	1002	0.0	{0.9555667741635393, 0.03312516553017035, 1.0806044881585473E-13, 0.003591770551602212, 9.667406400898496E-4, 9.284296325483018E-5, 8.023270783019094E-4, 8.312460016421437E-4, 1.5313993089205318E-6, 7.80094185137261E-4, 8.873525384011117E-4, 8.41741753547084E-4, 8.562600381784758E-4, 8.390339445965656E-4, 8.171192121220397E-4 }	

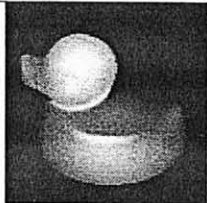
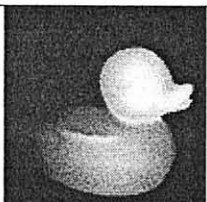
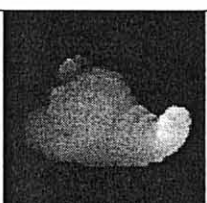
2	1001	0.026132039092972	<p>{0.9654401111888283, 0.011637771705267922, 5.682531717202108E-13, 0.004790180463411977, 0.01202152647769132, 6.391060358822784E-5, 7.416082036953665E-4, 7.501298801424712E-4, 7.540686695826432E-7, 7.160715753162513E-4, 7.985612486338762E-4, 7.572254016972766E-4, 7.805974592046016E-4, 7.515400490828037E-4, 7.500116742019378E-4 }</p>	
3	1009	0.035495796604414	<p>{0.9675570899997983, 0.004925922564622771, 1.7717176771032872E-12, 0.0034368315887537814, 0.018874342025991666, 3.255655290986277E-5, 6.398061065734357E-4, 6.421617189219539E-4, 1.0388387059115477E-6, 6.161079190494991E-4, 6.808922438087768E-4, 6.446019737744661E-4, 6.676099577384869E-4, 6.39665614080093E-4, 6.413728934988924E-4 }</p>	
19	5007	1.227273669343448	<p>{0.10267344851880422, 1.0470369434512749E-4, 1.491836970801549E-8, 0.004299340727588532, 0.8828292161384477, 2.7194210392781173E-6, 0.0012597868613646664, 0.0012356681526338059, 2.221234774079347E-4, 0.0012254262540945793, 0.0012556386776354273, 0.001222787062027765, 0.0012368648386134804, 0.0011917730767710946, 0.0012404881808568 }</p>	

TABELA 4: MINERAÇÃO DE IMAGENS COM REDES NEURAIIS

No quarto teste, foi usado o algoritmo EM, de aprendizado não supervisionado, configurado para criar 5 agrupamentos (classes). Possui uma margem de erro maior, pois atribui diferentes objetos à mesma classe, mas evidencia que não é obrigatório o uso de algoritmos supervisionados. Vide tabela a seguir:


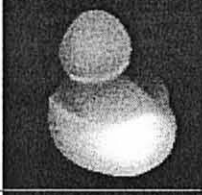
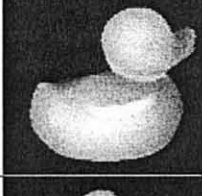
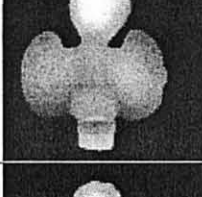

#	ID	Similaridade	Classificação	Imagem
1	1002	0.0	{ 1.0, 1.1308283367741167E-168, 5.171099081634771E-187, 3.789867890625547E-170, 1.1284713494188215E-54 }	
2	1016	1.1132520062362126E-54	{ 1.0, 3.884767506065973E-98, 3.977828791307398E-167, 2.960361974874167E-115, 1.5219343182608945E-56 }	
3	1011	1.1167580571461527E-54	{ 1.0, 1.3248589483617564E-105, 2.2569314057844426E-190, 1.557802799839392E-132, 1.1713292272668845E-56 }	
8	4010	1.1284713494188215E-54	{ 1.0, 2.7693653852901154E-67, 2.436677844344848E-217, 7.685289271409224E-75, 2.3522090848636747E-82 }	
11	4001	1.3853685893485227E-51	{1.0,2.6149078580249375E-71,0.0, 1.3853681297423907E-51, 6.320251773459075E-171 }	

TABELA 6: MINERAÇÃO DE IMAGENS COM O ALGORITMO EM

Apesar de ser um algoritmo de *Instance-based learning*, o algoritmo *KStar* (Tabela 7) apresentou uma performance melhor que o IBk nesse quinto teste, pois gera uma distribuição de probabilidade mais heterogênea.

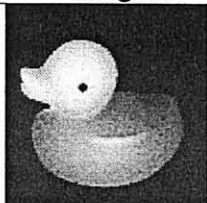
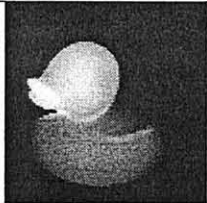


#	ID	Similaridade	Classificação	Imagem
1	1002	0.0	{0.99999999999916154, 8.87026829051846E-42, 5.262888600771541E-42, 8.384616777451173E-12, 1.903441144882415E-23, 3.314045595468051E-57, 0.0, 0.0, 1.130322022949036E-50, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 }	
2	1003	1.0305472636491E-11	{0.9999999999964201, 1.2762729147931892E-39, 1.1613416505796395E-39, 5.4171632864398275E-33, 3.5799034329139997E-12, 7.37255599731265E-110, 0.0, 0.0, 7.262900958226317E-35, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 }	
3	1011	1.0376344275770137E-11	{0.9999999999989526, 2.243667066104088E-45, 2.473616750685049E-44, 1.0474919004141589E-12, 4.028252952920931E-24, 1.1333353307413684E-62, 0.0, 0.0, 1.286481302414844E-55, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 }	
19	5007	1.4142133532234078	{1.4788696979026345E-7, 7.077375571970011E-39, 2.8988700738031054E-36, 4.6030663426424724E-58, 0.999998521130301, 2.583790556731447E-148, 0.0, 0.0, 3.53803227731671E-29, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 }	

TABELA 7: MINERAÇÃO DE IMAGENS COM O ALGORITMO K*

Finalmente, na Tabela 8, encontra-se o resultado da pesquisa combinando os 5 algoritmos, isto é, somando os valores de similaridades calculados para cada minerador (essa execução foi gerada independentemente das tabelas anteriores).

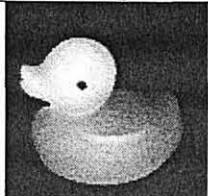
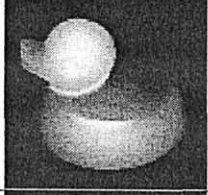
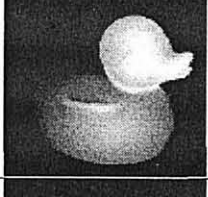
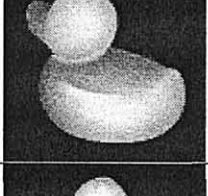
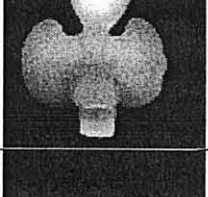
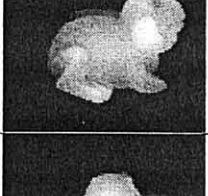

#	ID	Similaridade	Imagem
1	1002	0.0	
2	1001	0.0052264078209659694	
3	1009	0.007099159323254391	
4	1018	0.00746524253102802	
19	4010	1.0808631206929669	
31	2007	1.3268740306212787	
32	3009	1.330522439073666	

TABELA 8: SRIC COMBINANDO 5 ALGORITMOS DE MINERAÇÃO

Como consequência dos experimentos, é possível estabelecer um panorama do comportamento dos algoritmos, bem como, dos tipos de testes e evoluções que podem ser feitas no sistema. No caso dos algoritmos, observou-se que é imprescindível a possibilidade de alterar a influência de cada algoritmo na pesquisa, seja atribuindo um peso para cada um ou normalizando as distribuições de probabilidade. Na primeira alternativa, se o usuário desejar buscar imagens que estão no mesmo conceito, deve ser aumentada a relevância de algoritmos como PART e IBk; se o usuário deseja imagens com características semelhantes, deve priorizar redes neurais e clusterizadores, por exemplo. A segunda alternativa se refere ao fato de que algoritmos como *KStar* ou EM calculam valores de probabilidade muito pequenos, ao passo que redes neurais, PART e IBk geram valores mais altos.

O próximo passo é utilizar ferramentas e metodologias adequadas de teste. Nessa situação, que envolve tanto Mineração de Dados quanto CBIR, a gama de opções e a complexidade é ainda maior. No que se refere à CBIR, conforme [Smeulders et al. 00], podem ser usados conceitos como precisão (fração de imagens relevantes da consulta sobre o total de imagens retornada) e revocação (*recall* ou fração de imagens relevantes retornadas pela consulta sobre o total de imagens relevantes). Entretanto, o próprio Smeulders afirma que no caso de imagens, a definição de relevância é subjetiva. Tratando-se de Mineração de Dados (e Imagens), [Witten & Frank 00] apontam vários testes, inclusive cálculo do custo do algoritmo (quando gera classificações erradas), cálculo de taxas de erro (tais como erro médio, erro absoluto, erro relativo e coeficiente de correlação) e *cross-validation* (validação do modelo utilizando diferentes partições de dados).

Por meio das avaliações e experiências descritas nesta Seção, observa-se que o tema não está esgotado. Além dos testes e da melhoria do critério de similaridade, é importante continuar a pesquisa de modo a avaliar a performance do sistema com bases de dados maiores. Essa constatação abre espaço para novos estudos na área de Banco de Dados (indexação), Mineração de Dados (adaptação de algoritmos de mineração para o campo de imagens) e Visão Computacional (representação de características e algoritmos de extração).

9 CONCLUSÃO E TRABALHOS FUTUROS

9.1 CONCLUSÃO

Ao longo desse trabalho foram apresentados os conceitos, desafios e exemplos de uso da Mineração de Imagens. Como uma tecnologia multidisciplinar, a Mineração de Imagens toma emprestado algoritmos, técnicas e metodologias de outras disciplinas, principalmente Mineração de Dados, Visão Computacional, SRIC e Bancos de Dados. Mais do que uma fusão de soluções, a Mineração de Imagens vem endereçar problemas específicos do domínio de dados visuais, tais como a escolha da melhor modelagem do banco de dados para representação das características extraídas e do conhecimento gerado, ou a adaptação dos algoritmos de Mineração de Dados tradicional para o uso em base de imagens.

De maneira semelhante às fases do processo de KDD, também foi proposto um conjunto de etapas que devem compor um sistema de Mineração de Imagens. Essas etapas incluem desde a aquisição de imagens, a extração de características, a indexação e modelagem do banco de dados, a descoberta de conhecimento propriamente dita até o refinamento, validação e testes. Um dos diferenciais da Mineração de Imagens é a necessidade de se trabalhar com vários níveis de informação, tal como em sistemas de Visão Computacional e em SRIC. Isso obriga a existência de uma atividade de transformação de informações – baixo nível (cor, *pixel*), nível intermediário (forma, textura, momentos, dentre outros) e alto nível (conhecimento) - incorporada ao processo de KDD.

De posse dos dados visuais, isto é, as características da imagem, e textuais, tais como dados médicos ou geográficos e campos descritivos da imagem, é necessário escolher a estratégia da mineração. Se o objetivo é prever o valor de um atributo de modo a classificar uma imagem, devem ser escolhidos algoritmos de

classificação. Se o objetivo é conhecer os dados, procurando extrair conhecimento implícito, o usuário poderá aplicar algoritmos de descoberta de relações ou de agrupamento. Outras tarefas também podem ser desempenhadas e, na prática, um mesmo algoritmo pode ser aplicado para uma ou mais estratégias. Além da estratégia utilizada, o que guia a escolha de um algoritmo é o tipo de dado que um sistema irá trabalhar e outros fatores tais como desempenho, precisão e a legibilidade do conhecimento a ser gerado. Como não existe uma solução perfeita, um sistema de Mineração de Imagens deve ser flexível a ponto do usuário poder escolher com base em experiências qual o melhor algoritmo a ser utilizado. Com isso, o usuário poderá substituir redes Bayesianas por redes neurais, substituir *rough sets* por *lógica fuzzy*, ou combinar várias técnicas.

As abordagens para se extrair conhecimento a partir de dados de baixo nível e nível intermediário também motivam a classificação das arquiteturas de sistemas de Mineração de Imagens. Conforme proposto, um sistema pode ser orientado a modelo (já existe um conhecimento prévio que precisa ser validado), orientado a dados (um modelo está sendo construído), orientado à função (com módulos que seguem as etapas de um processo de KDD) e orientado à informação (com foco na estrutura dos dados). Outra classificação trata do tipo de aplicação da mineração: como um sistema único, como um auxiliar a processamentos de baixo nível ou como integrante de um SRIC.

Além da apresentação de conceitos, do processo e de exemplos de Mineração de Imagens, outra contribuição desse trabalho foi a implementação de um sistema de Mineração de Imagens aplicado à consulta e classificação de imagens de profundidade. O módulo de mineração, independente do SRIC, permite a configuração dos parâmetros que serão usados para a mineração das imagens (escolha das características e dos algoritmos de aprendizado de máquina), a geração de uma base de conhecimento e o posterior uso dessa base em outros sistemas – no caso, para recuperação de imagens por conteúdo.

Em resumo, a Mineração de Imagens abre inúmeras possibilidades de utilização e pesquisa. Mesmo no caso de imagens de profundidade, a solução desenvolvida ainda poderá ser usada como auxiliar na reconstrução de modelos 3D a partir das vistas dos objetos. Como temas de pesquisa, novos algoritmos e

metodologias podem ser desenvolvidos ou adaptados para o domínio de imagens, tais como algoritmos de aprendizado de máquina capazes de trabalhar com características mais complexas ou ainda esquemas de indexação voltados para a representação de conhecimento. O uso da Mineração de Imagens também pode ser expandida para outras áreas, tais como medicina e sensoriamento remoto, integrando não apenas imagens, como também dados textuais e multimídia. Enfim, uma nova tecnologia está disponível, permitindo transformar dados brutos obtidos de imagens em conhecimento útil, com possibilidade de aplicação em várias áreas do conhecimento.

9.2 TRABALHOS FUTUROS

Como consequência da arquitetura proposta e dos experimentos realizados, vários tipos de avanços podem ser feitos dentro do contexto de Mineração de Imagens, SRIC e imagens de profundidade. Assim, é possível visualizar trabalhos futuros que envolvem tanto a melhoria da arquitetura do Sistema de Mineração de Imagens, como também desenvolvimento de novos algoritmos e metodologias ou o estudo de novas aplicações do tema.

Um sistema de Mineração envolve dois aspectos chave: a extração de características e a mineração propriamente dita. Quanto à extração, o grande desafio está em como representar e modelar corretamente as características das imagens, que normalmente extrapolam a noção de dados numéricos ou caracter. Quanto à mineração, a dificuldade reside em como adaptar os algoritmos tradicionais para entender esses tipos dados, bem como, prover implementações que tenham um desempenho aceitável de execução.

Quanto aos tipos de dados, tomando apenas como referência o SRIC3D, é possível enumerar a seguinte variedade de características:

- a) características de textura e momentos: valores numéricos, que podem ser modelados facilmente em campos numéricos num banco de dados;

- b) descritores de superfície: nesse caso, a quantidade de atributos utilizada é variável, isto é, um CSS e descritores de Fourier possuem um número variável de coeficientes e, no caso de Fourier, a ordem dos mesmos é importante;
- c) regiões: as mesmas informações de textura, momentos e descritores de superfície são extraídas por região de uma imagem, gerando uma quantidade variável de regiões;
- d) informação espacial: essas mesmas regiões também podem ser processadas para dar origem a atributos que representam relações espaciais, inclusive *2DString*.

Se consideradas as aplicações na área médica ou de sensoriamento remoto, as mesmas características acima devem ser combinadas de maneira eficiente junto com dados textuais (diagnósticos, classificações de terreno) e multimídia (documentos e imagens de outras naturezas, como imagens de intensidade, tomografia, raio-x, etc...). Como resultado, pesquisas na área de bases multidimensionais, bases espaciais e bases multimídia são imprescindíveis para se chegar a um nível de modelagem satisfatório. Naturalmente, os algoritmos atuais precisam ser modificados para compreender o significado de tipos de dados mais complexos, porém sem desconsiderar o aspecto de desempenho.

Como observado na visão dinâmica (Seção 8.3.2) da arquitetura do sistema, a performance do algoritmo de mineração é um fator importante para a viabilidade do aplicativo. Dessa maneira, a execução do algoritmo deve ser rápida, preferencialmente o modelo deve ser incremental (isto é, precisa ser atualizado facilmente na presença de uma nova instância) e permitir sua persistência (gravação e recuperação no banco) para evitar que todo o modelo seja refeito durante a reinicialização do sistema. Assim, além da pesquisa na área de Aprendizado de Máquina e Mineração de Dados, é importante o desenvolvimento de estruturas e bancos de dados capazes de representar corretamente o conhecimento (modelo, regras, árvores) gerado pela mineração.

Tomando como ponto de partida o aplicativo desenvolvido, novos estudos na área de Processamento de Imagens e Visão Computacional podem ser feitos,

principalmente no que se refere à extração de características. Além da já citada reconstrução de objetos 3D a partir das vistas, alguns tópicos podem ser trabalhados:

- a) pesquisa de novas características e esquemas de representação (como *2DString*, CSS, descritores de Fourier);
- b) avaliação e seleção das características mais relevantes;
- c) testes do sistema com imagens que possuem mais de um objeto ou que possuam ambientes mais complexos (cenários externos, imagens ruidosas);
- d) combinação de características de imagens de intensidade e imagens de profundidade.

No que se refere a Sistemas de Recuperação de Imagens por Conteúdo, esse trabalho já evidenciou a importância da incorporação do conhecimento como ferramenta alternativa (que foi o caso do aplicativo desenvolvido) ou complementar para as atuais abordagens de sistemas de busca. Entretanto, o uso da mineração acarreta a necessidade de indexar o conhecimento gerado, motivando o estudo de mecanismos de indexação adequados para os vetores de características gerados (como a distribuição de probabilidade).

A aplicação da Mineração de Imagens não se restringe a SRIC, mas também a sistemas de outros domínios, como segue:

- a) tutores inteligentes e sistemas especialistas, atuando como componente que evolui o modelo baseado no cadastro de imagens e na realimentação do próprio usuário;
- b) na área médica, como auxiliar a sistemas de diagnóstico;
- c) na área de sensoriamento remoto, para ajudar a classificação e fusão de imagens ;
- d) na área de Visão Computacional com robótica, como um complemento a sistemas baseados em modelos, uma vez que uma das características mais interessantes da mineração de imagens é o fator de aprendizado.

A exploração de algoritmos de aprendizado de máquina com imagens não é recente na Visão Computacional, mas o uso intensivo da tecnologia aliada a banco de dados é uma área recente que abre diversas possibilidades de expansão. Um dos aspectos fundamentais da importância da Mineração de Imagens está na combinação do crescente número de características que podem ser extraídas das imagens (frutos de pesquisas nas áreas de segmentação e representação) com o aumento do poder de processamento e armazenamento dos computadores atuais.

10 REFERÊNCIAS BIBLIOGRÁFICAS

- [Abiteboul 01] Abiteboul, S. Semistructured data: from practice to theory. *Proceedings of 16th Annual IEEE Symposium on Logic in Computer Science*. p.379-386, EUA, 2001.
- [Adriaans & Zantinge 96] Adriaans, P.; Zantinge, D. *Data mining*. Syllogic, 1996.
- [Aha & Kibler 91] Aha, D.; Kibler, D. Instance-based learning algorithms. *Machine Learning*. v.6 p.37-66, 1991
- [Althoff et al 98] Althoff, K. D; Bergmann, R.; Wess, S.; Manago, M. Case-based reasoning for medical decision support tasks: the Inreca approach. *Artificial Intelligence in Medicine*. v.12, p.25-41,1998.
- [Araújo & Guimarães 00] Araújo, A.A.; Guimarães, S.J.F. Recuperação de informação visual com base no conteúdo em imagens e vídeos digitais. *Revista de Informática Teórica e Aplicada – RITA*. UFRGS, Brasil, 2000.
- [Azevedo et al. 00] Azevedo, F.M.; Brasil, L.M; Oliveira, R.C.L. *Redes neurais com aplicações em controle e em sistemas especialistas*. Visual Books, 2000.
- [Azuaje et al. 99] Azuaje, F.; Dubitzky, W.; Lopes, P.; Black, N.; Adamson, K.; Wu, X. White, J.A. Predicting coronary disease risk based on short-term RR interval measurements: a neural network approach. *Artificial Intelligence in Medicine*. v.15, p.275-297, 1999.
- [Bellon et al. 99a] Bellon, O.R.P.; Direne, A.; Silva, L. Edge detection to guide range image segmentation by clustering techniques. *Proceedings of the 6th IEEE International Conference on Image Processing*. v.2, p.725–729, Japão, 1999.
- [Bellon et al. 99b] Bellon, O.R.P.; Direne, A.I.; Silva, L. Building meaningful edge maps from range images *Proceedings of the 10th IAPR International Conference on Image Analysis and Processing*. Itália, 1999.
- [Bellon et al. 02] Bellon, O.R.P.; Gorga, C.; Cat, M.; Silva, L.; Severich, M. Um sistema de recuperação de imagens para apoiar a determinação da idade gestacional de recém nascidos. *XXIX Seminário Integrado de Software e Hardware*. p.193-202, Florianópolis-SC, 2002.
- [Bellon & Silva 02] Bellon, O.R.P.; Silva, L. New improvements on range image segmentation by edge detection techniques. *IEEE Signal Processing Letters*. v.9, n.2, p.43-45, 2002.

- [Bellon et al. 01] Bellon, O.R.P.; Silva, L.; Gotardo, P. Edge-based segmentation using curvature sign maps of reflectance and range images. *Proceedings of 8th IEEE International Conference on Image Processing – ICIP Thessaloniki-Greece*. v.1, p.730–734, Grécia, 2001.
- [Besl 88] Besl, P.J. Active optical range imaging sensors. *Advances in machine vision*. Springer-Verlag, 1988.
- [Bozkaya & Ozsoyoglu 97] Bozkaya, T.; Ozsoyoglu, M. Distance-based indexing for high-dimensional metric spaces. *Proceedings of SIGMOD International Conference of Management of Data*. p. 357-368, 1997.
- [Brooks 81] Brooks, R. Symbolic reasoning among 3-D models and 2-D images. *Artificial Intelligence* 17. p.285-348, 1981.
- [Burl et al. 99] Burl, M. C.; Fowlkes, C.; Roden, J. Mining for image content. *Systemics, Cybernetics and Informatics: Information Systems Analysis and Synthesis*. EUA, 1999.
- [Carey et al. 97] Carey, R.; Bell, G.; Marrin, C. *ISO/IEC 14772-1:1997 - Virtual Reality Modeling Language*. VRML Consortium, 1997.
- [Carston et. al 99] Carson, C.; Belongie, S.; Thomas, M. Blobworld: a system for region-based image indexing and retrieval. *Third International Conference on Visual Information Systems*. 1999.
- [Carvalho 01] Carvalho, L. *Datamining: a mineração de dados no marketing, medicina, economia, engenharia e administração*. Érica, 2001.
- [Celinski 98] Celinski, T.M. Métodos de agrupamento: uma abordagem comparativa com aplicação em segmentação de imagens de profundidade. *Dissertação de Mestrado*, UFPR. 1998.
- [Celinski & Bellon 98] Celinski, T.M.; Bellon, O.R.P. Métodos de agrupamento. *Tutorial XI SIBPRAPI*. 1998.
- [Cleary & Trigg 95] Cleary, J.G.; Trigg, L.E. K*: an instance-based learner using and entropic distance measure. *Proceedings of the 12th International Conference on Machine learning*. p.108-114. 1995.
- [Croquette et al. 02] Croquette, O.; Vandeborre, J.P.; Daoudi, M; Chaillou, C. Indexing and retrieval VRML models. *SPIE Electronic Imaging 2002 Symposium*. EUA, 2002.
- [Datcu & Seidel 00] Datcu, M.; Seidel, K. Image Information Mining: Exploration of Image Content in Large Archives. *IEEE Aerospace Conference Proceedings*. EUA, 2000.
- [Del Bimbo 99] Del Bimbo, A. *Visual information retrieval*. Morgan Kaufmann, 1999.

- [Dempster et al. 77] Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*. 1ª ed. v. 39. p. 1-38. 1977.
- [Direne 97] Direne, A. I. Authoring intelligent systems for teaching visual concepts. *International Journal of Artificial Intelligence in Education (IJAIED)*. v.8, n.1, p.44-70, 1997.
- [Djeraba 01] Djeraba, C. Relationship extraction from large image databases. *Proceedings of the 2nd International Workshop on Multimedia Data Mining*. p. 44-49, EUA, 2001.
- [Djeraba 00] Djeraba, C. When image indexing meets knowledge discovery. *Proceedings of the First International Workshop on Multimedia Data Mining*. p. 73-81, EUA, 2000.
- [Downing & Clark 00] Downing, D.; Clark, J. *Estatística aplicada*. Saraiva, 2000.
- [Dzeroski 96] Dzeroski, S. Inductive logic programming and knowledge discovery in databases. *Advances in Knowledge Discovery and Data Mining*. p.117-152, AAAIPress, 1996.
- [Eakins 02] Eakins, J.P. Towards intelligent image retrieval. *Pattern Recognition*. v.35, p.3-14, 2002.
- [Fayyad et al. 96a] Fayyad, U.; Djorgovski, S.G.; Weir, N. Automating the analysis and cataloging of sky surveys. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*. p.473-493, 1996.
- [Fayyad et al. 96b] Fayyad, U.; Shapiro, G.; Smyth, P. Knowledge discovery and data mining: towards a unifying framework. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* p.1-34, 1996.
- [Fisher 89] Fisher, R. B. *From surfaces to objects: computer vision and three dimensional scene analysis*. John Wiley & Sons, 1989.
- [Freitas & Lavington 98] Freitas, A.; Lavington, S. H. *Mining very large databases with parallel processing*. Kluwer, 1998.
- [Gaede & Gunther 98] Gaede, V.; Gunther, O. Multidimensional Access Methods. *ACM Computing Surveys*. 1998.
- [Garner 95] Garner, S.R. WEKA: The Waikato Environment for Knowledge Analysis. *Proceedings of New Zealand Computer Science Research Students Conference*. Nova Zelândia, 1995.
- [Gonzales & Woods 92] Gonzales, R.C.; Woods, R.E. *Processamento de imagens digitais*. Edgard Blücher, 1992.

- [Gorga et al. 02] Gorga, C.N.; Bellon, O.R.P.; Cat, M.; Sunye, M.; Marchaukoski, J. N.; Silva, L. A health care information system to neonatology support. *Proceedings of the 15th IEEE International Conference on Computer Based Medical Systems*. p.29-34, Maribor-Eslovênia, 2002.
- [Gotardo 02] Gotardo, P.F.U. Segmentação de imagens de profundidade utilizando curvaturas de superfície e um método de estimativa robusto. *Dissertação de Mestrado*, UFPR. 2002.
- [Gudivada & Jung 95] Gudivada, V.N; Jung, G.S. Spatial knowledge representation and retrieval in 3-D image databases. *Proceedings of IEEE International Conference on Multimedia Computing and Systems*. p. 90-97, 1995.
- [Gudivada & Raghavan 95] Gudivada, V.N; Raghavan, V.V. Content-based image retrieval systems. *IEEE Computer*. p. 18-22, 1995.
- [Han et al. 97] Han, J.; Chiang, S.; Chen, J.; Chen, Q.; Gong, W.; Kamber, M; Liu, G.; Koperski, K.; Lu, Y.; Stefanovic, N.; Winstone, L.; Xia, B.; Zaiane, O.R.; Zhang, S.; Zhu, H. DBMiner: a system for data mining in relational databases and datawarehouses. *Proceedings of CASCON'97 Meeting of Minds*. p. 8-13, Canada, 1997.
- [Haralick et al. 95] Haralick, R.M.; Shanmugam, K.; Dinstein, I. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*. v.3, n.6, p. 610-621, 1973 .
- [Hasan & Hyland 01] Hasan, H.; Hyland, P. Using OLAP and multidimensional data for decision making. *IT Professional*. v.3, n.5, p. 44-50, 2001.
- [Hayashi et al. 00] Hayashi, Y.; Setiono, R.; Katsumi, Y. A comparison between two neural network rule extraction techniques for the diagnosis of hepatobiliary disorders. *Artificial Intelligence in Medicine*. v.12, p. 205-216, Elsevier, 2000.
- [Heckerman 96] Heckerman, D. Bayesian networks for knowledge discovery. *Advances in knowledge discovery and data mining*. p. 273-304, AAAIPress, 1996.
- [Hipp et al. 00] Hipp, J.; Güntzer, U. Nakhaeizadeh, G. Algorithms for association rule mining—a general survey and comparison. *ACM SIGKDD Explorations*. v.2, n.1, p.58-63, 2000.
- [Hoover 98] Hoover, A. The space envelope representation for 3D scenes. *Computer Vision and Image Understanding* . 1998. Disponível em: <http://vision.ucsd.edu/~hoover/>.
- [Jain et al. 95] Jain, R.; Kasturi, R.; Schunck, B.G. *Machine vision*. Mcgraw-Hill, 1995.

- [Kakimoto et al. 00] Kakimoto, M.; Morita, C.; Tsukimoto, H. Data mining from functional brain images. *Proceedings of the First International Workshop on Multimedia Data Mining*. p. 91-97, EUA, 2000.
- [Kawahara & Kawano 01] Kawahara, N.; Kawano, H. Mining association algorithm with threshold based on ROC analysis. *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*. EUA, 2001.
- [Khoshafian & Barei 96] Khoshafian, S.; Barei, A.B. *Multimedia and image databases*. Morgan Kaufmann, 1996.
- [Kim 95] Kim, W. *Modern database systems: the object model, interoperability and beyond*. Addison Wesley, 1995.
- [Komorowski et al. 98] Komorowski, J.; Polwoski, L.; Skowron, A. *Rough sets: a tutorial*. Springer-Verlag, 1998.
- [Koperski & Marchisio 00] Koperski, K.; Marchisio, G. B. Multi-level indexing and GIS enhanced learning for satellite imageries. *Proceedings of the First International Workshop on Multimedia Data Mining*. EUA, 2000.
- [Korth & Silberschatz 95] Korth, H.; Silberschatz, A. *Sistemas de bancos de dados*. Makron Books, 1995.
- [Lavrac 99] Lavrac, Nada. Selected techniques for data mining in medicine. *Artificial Intelligence in Medicine*. v.16, p.3-23, Elsevier, 1999.
- [Liang & Kuo 99] Liang, K.; Kuo, C.C.J. WaveGuide: a joint wavelet-based image representation and description system. *IEEE Transactions on Image Processing*. v.8, n.11, p.1619-1629, 1999.
- [Mahmoudi & Dadoudi 02] Mahmoudi, S.; Dadoudi, M. 3D models retrieval and indexing. *Proc. 6th Joint Conference on Information Sciences*. EUA, 2002.
- [Masulli & Schenone 99] Masulli, F.; Schenone, A. A fuzzy clustering based segmentation system as support to diagnosis in medical imaging. *Artificial intelligence in medicine*. v.6, p.129-147, 1999.
- [Mitchel 97] Mitchel, T. M. *Machine learning*. McGraw-Hill, 1997.
- [Mokhtarian et al. 96] Mokhtarian, F.; Abbasi, S.; Kittler, J. Efficient and Robust Shape Indexing through Curvature Scale Space. *Proceedings of British Machine Vision Conference*. p. 53-62. Inglaterra, 1996.
- [Nadler & Smith 93] Nadler, M.; Smith, E. *Pattern recognition engineering*. John Wiley & Sons, 1993.
- [Nascimento 02] Nascimento, J. P. Análise e classificação de imagens utilizando características de textura. *Dissertação de Mestrado*, UFPR. 2002.

- [Øhrn et al. 98] Øhrn A.; Komorowski J.; Skowron A.; Synak, P. The design and implementation of a knowledge discovery toolkit based on rough sets: the Rosetta system. *Studies In Fuzziness And Soft Computing*. v.8, Physica-Verlag, 1998.
- [Ordonez & Omiecinski 99] Ordonez, C.; Omiecinski, E. Discovering association rules based on image content. *Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries Conference (ADL'99)*. p. 38-49. EUA, 1999.
- [Pal 92] Pal, S.K. Fuzzy Sets in Image Processing and Recognition. *IEEE International Conference on Fuzzy Systems*. p.119-126, 1992.
- [Pedersen & Jensen 01] Pedersen, T.B.; Jensen, C.S. Multidimensional database technology. *IEEE Computer*. v. 34, n.12, p.40-46, 2001.
- [Peña & Sipper 99] Peña, C.A.; Sipper, M. Evolutionary computation in medicine. *Artificial Intelligence in Medicine*. v.17, p.131-155, Elsevier, 1999.
- [Pereira 01] Pereira, S.V. Classificação e avaliação de Sistemas de Recuperação de Imagens por Conteúdo. *Dissertação de Mestrado, UFPR*. 2001.
- [Rui et al. 99] Rui, Y.; Huang, T.S.; Chang, S. Image retrieval: current techniques, promising directions and open issues. *Journal Visual Communications And Image Representation*. v.10, p. 39-62, 1999.
- [Samet 95] Samet, H. Spatial data structures. *Modern database systems*. Addison-Wesley, 1995.
- [Sedgewick 98] Sedgewick, R. *Algorithms in C*. Addison-Wesley, 1998.
- [Shih et al. 02] Shih, T. K.; Hung, Y; Huang, J.; Simeon, J. Content-based information retrieval in large VR scene database. *Proceedings of Joint conference on Information Sciences*. EUA, 2002.
- [Silva 03] Silva, L. Segmentação e modelagem de objetos 3D a partir de várias vistas de imagens de profundidade. *Tese de Doutorado, CEFET-PR*. 2003.
- [Silva 02] Silva, L.; Bellon, O.R.P.; Gotardo, P. A global-to-local approach for robust range image segmentation. *Proceedings of the 9th IEEE International Conference on Image Processing (ICIP2002)*. EUA, 2002.
- [Siromoney et al. 00] Siromoney, A.; Raghuram, L.; Korah, A.; Prasad, G.N.S. Inductive logic programming for knowledge discovery from MRI data. *IEEE Engineering in Medicine and Biology Magazine*. v.19, p.72-77, 2000.
- [Smeulders et al. 00] Smeulders, Arnold W.M.; Worring, M.; Santini, S. Content-based image retrieval at the end of the early years. *IEEE Transactions on PAMI*. v.22, n.12, p.1349-1380, 2000.

- [Smyth et al. 96] Smyth, P.; Fayyad, U.M.; Burl, M.C.; Peronna, P. Modeling subjective uncertainty in image annotation. *Proc. Int. Conf. on Knowledge Disc. and Data Mining*. p.717-739, 1996.
- [Soh & Tsatsoulis 99] Soh, L.; Tsatsoulis, C. Segmentation of satellite imagery of natural scenes using data mining. *IEEE Transactions on Geoscience and Remote Sensing*. v.37, n.2, p.1086-1099, 1999.
- [Sonka et al. 99] Sonka, M.; Hlavac, V.; Boyle, R. *Image processing, analysis and machine vision*. PWS Publishing, 1999.
- [Soodamani & Liu 00] Soodamani, R.; Liu, Z.Q. GA-based learning for a model-based object recognition system. *International Journal of Approximate Reasoning*. v.23, p.85-109, Elsevier, 2000.
- [Steimann 97] Steimann, F. Fuzzy set theory in medicine. *Artificial Intelligence in Medicine*. v.11, p.1-7, 1997.
- [Suk & Bhandarkar 92] Suk, M.; Bhandarkar, S. M. *Three dimensional object recognition from range images*. Springer Verlag, 1992.
- [Tanomaru 95] Tanomaru, J. Motivação, fundamentos e aplicações de algoritmos genéticos. *II Congresso Brasileiro de Redes Neurais - III Escola de Redes Neurais*. Curitiba, 1995.
- [Vailaya 00] Vailaya, A. *Semantic classification in image databases*. Teste de Doutorado, Department of Computer Science, Michigan State University, 2000.
- [Vailaya et al. 96] Vailaya, A.; Zhong, Y.; Jain, A.K. A hierarchical system for efficient image retrieval. *Proceedings of the 13th Conference on Pattern Recognition*. Áustria, 1996.
- [Vieira 02] Vieira, E. V.; Bellon, O.R.P.; Silva, L. Mineração de Imagens. *Revista de Informática Teórica Aplicada - RITA*. v.9, n.2, p.67-96. UFRGS, Brasil, 2002.
- [Wang & Chung 01] Wang, Q.; Chung, S.M. Content-based retrieval and data mining of a skin cancer image database. *Proceedings of the International Conference on Information Technology: Coding and Comp*. p.611-615, EUA, 2001.
- [Weschler 90] Weschler, H. *Computational vision*. Academic Press, 1990.
- [Witten & Frank 98] Frank, E.; Witten, I.H. Generating accurate rule sets without global optimization. *Machine Learning: Proceedings of the 15th International Conference*. Morgan Kaufmann, 1998.
- [Witten & Frank 00] Witten, I.H.; Frank E.B. *Data mining: practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann, 2000.
- [Zadeh 65] Zadeh, L.A. Fuzzy Sets. *Information And Control*. p. 338-353, 1965.

- [Zaiane & Han 00] Zaiane, O.R.; Han, J. Discovering Spatial Associations. *Images Data Mining and Knowledge Discovery: Theory, Tools, and Technology II, SPIE 14th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*. EUA, 2000.
- [Zaiane et al. 98] Zaiane, O.R.; Han, J.; LI, Z. Multimedia-miner: a system prototype for multimedia data mining. *ACM-SIGMOD Conference on Management Data*. 1998.
- [Zaiane et al. 00] Zaiane, O.R.; Han, J.; Zhu, Hua. Mining Recurrent Items in Multimedia with Progressive Resolution Refinement. *Proceedings of the 16th International Conference on Data Engineering (ICDE2000)*. p. 461-470. EUA, 2000.
- [Zhang et al. 01] Zhang, J.; Hsu, W.; Lee, M. L. Image mining: issues, frameworks and techniques. *Proceedings of the 2nd International Workshop on Multimedia Data Mining*. p. 13-20. EUA, 2001.