Ambiente de Gerenciamento de Imagens e Dados Espaciais para Desenvolvimento de Aplicações em Biodiversidade

Ricardo da Silva Torres

Tese de Doutorado

Instituto de Computação
Universidade Estadual de Campinas

Ambiente de Gerenciamento de Imagens e Dados Espaciais para Desenvolvimento de Aplicações em Biodiversidade

Ricardo da Silva Torres[1]

15 de outubro de 2004

Banca Examinadora:

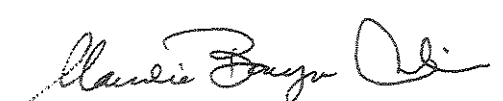
- Prof. Dra. Claudia Bauzer Medeiros (Orientadora)
- Profa. Dra. Ana Carolina Brandão Salgado
  CIN - UFPE
- Profa. Dra. Agma Juci Machado Traina
  ICMC - USP
- Prof. Dr. Alberto Henrique Frade Laender
  DCC - UFMG
- Prof. Dr. Neucimar Jerônimo Leite
  IC - UNICAMP

[1]**Suporte financeiro da FAPESP (Procs. 00/04872-2 e 01/02788-7), CAPES (Proc. BEX844/03-9) e apoio parcial dos projetos SAI/PRONEX-MCT e Web-Maps do CNPq.**

Ambiente de Gerenciamento de Imagens e Dados Espaciais para Desenvolvimento de Aplicações em Biodiversidade

Este exemplar corresponde à redação final da Tese devidamente corrigida e defendida por Ricardo da Silva Torres e aprovada pela Banca Examinadora.

Campinas, 15 de outubro de 2004.



Prof. Dra. Claudia Bauzer Medeiros
(Orientadora)

Tese apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Torres, Ricardo da Silva

T636a Ambiente de gerenciamento de imagens e dados espaciais para desenvolvimento de aplicações em biodiversidade / Ricardo da Silva Torres – Campinas, [S.P. :s.n.], 2004.

Orientadora : Claudia Maria Bauzer Medeiros

Co-Orientador: Alexandre Xavier Falcão

Tese (doutorado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Sistemas de recuperação da informação. 2. Biodiversidade. 3. Bibliotecas digitais. 4. Processamento de imagens. I. Medeiros, Claudia Maria Bauzer. II. Falcão, Alexandre Xavier. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

TERMO DE APROVAÇÃO

Tese defendida e aprovada em 15 de outubro de 2004, pela Banca examinadora composta pelos Professores Doutores:

Profa. Dra. Ana Carolina Brandão Salgado
UFPE

Profa. Dra. Agma Juci Machado Traina
ICMC - USP

Prof. Dr. Alberto Henrique Frade Laender
UFMG

Prof. Dr. Neucimar Jerômino Leite
IC - UNICAMP

Profa. Dra. Claudia Maria Bauzer Medeiros
IC - UNICAMP

"A maior recompensa do nosso trabalho não é o que nos pagam por ele, mas aquilo em que ele nos transforma."

(*John Ruskin*)

Resumo

Há um grande número de aplicações ambientais requisitando o gerenciamento sofisticado de vários tipos de dados, incluindo dados espaciais e imagens de seres vivos. Entretanto, os sistemas de informação disponíveis oferecem suporte limitado para gerenciamento destes dados de uma maneira integrada. Por um lado, aplicações ambientais baseadas em Sistemas de Informação Geográfica permitem a correlação espacial de dados geofísicos e informação de espécies vivas. Por outro lado, sistemas de informação de imagens usados por biólogos suportam o gerenciamento de fotos de paisagens e/ou animais, mas sem nenhum tipo de referência espacial. Esta tese provê uma solução que combina estes requisitos de consultas, aproveitando-se da tecnologia de bibliotecas digitais para gerenciar coleções de dados heterogêneos de maneira integrada. Esta pesquisa contribui, desta forma, para resolver problemas de especificação e implementação de sistemas de informação de biodiversidade que combinem o gerenciamento de imagens de seres vivos, descrições textuais e dados espaciais no contexto de bibliotecas digitais. Esta solução provê pesquisadores em biodiversidade com novas opções de consulta.

As principais contribuições desta tese são: (i) uma arquitetura genérica, baseada em componentes de bibliotecas digitais, para gerenciamento de coleções de dados heterogêneos, para acessar fontes de dados de biodiversidade (texto, imagens e dados espaciais); (ii) a proposta de novos descritores de forma para suportar a recuperação de imagens por conteúdo; (iii) um novo componente de biblioteca digital para busca de imagem por conteúdo; (iv) adoção de estruturas visuais distintas para exploração de resultados em bancos de dados de imagens; e (v) validação parcial da arquitetura, através de um protótipo que usa dados sobre peixes.

Abstract

There is a wide range of environmental applications requiring sophisticated management of several kinds of data, including spatial data and images of living beings. However, available information systems offer very limited support for managing such data in an integrated manner. On the one hand, environmental applications based on Geographic Information Systems (GIS) allow spatially correlating geophysical data and information on living species. On the other hand, image information systems used by biologists provide management of photos of landscapes and/or animals, but without any kind of geographical referencing. This thesis provides a solution to combine these query requirements, which takes advantage of current digital library technology to manage networked collections of heterogeneous data in an integrated fashion. The research thus contributes to solve problems of specification and implementation of biodiversity information systems that manage images of species, textual descriptions and spatial data in an integrated way, under the digital library perspective. This solution provides biodiversity researchers with new querying options.

The main contributions of this thesis are: (i) a generic architecture, based on digital library components, for managing heterogeneous data collections, to access biodiversity data sources (text, images, and spatial data); (ii) a proposal of new shape descriptors for supporting content-based image retrieval; (iii) a new digital library component, for content-based image search; (iv) adoption of distinct visual structures for exploring query results in an image database; and (v) partial validation of the architecture, through implementation of a prototype that uses fish-related data.

Agradecimentos

Agradeço...

A Deus, por minha vida.

À minha esposa Moema. Este trabalho não seria possível sem o seu amor, carinho, incentivo e compreensão.

Aos meus pais, Raimundo (*Painho*) e Auria (*Mainha*), e aos meus irmãos (Angela, Ronaldo, Marilia e Egnaldo). Agradeço por todo amor e carinho. Obrigado pelos conselhos e exemplos de determinação.

À Profa. Claudia Bauzer Medeiros, pelo estímulo, paciência e, acima de tudo, pelos conselhos e críticas. Agradeço por haver proporcionado tantas oportunidades para o meu crescimento intelectual e emocional.

Ao Prof. Alexandre Xavier Falcão, pelo exemplo de dedicação e empenho. Agradeço pelo constante encorajamento e apoio dado durante a realização deste trabalho.

Ao Prof. Edward Fox, pela atenção e estímulo durante minha estadia na Virginia Tech.

Aos amigos com quem sempre contarei: Dna. Zelia e Seu Miguel, Seu Toninho e Dna. Maria, Osmar, Unni, Ramesh, Paulo, Mauricio, Tooru, Walid, Leonore, Sindi... Todos, direta ou indiretamente, têm me ensinado muito... fazendo minha vida mais simples e bela.

Aos colegas da Unicamp e Virginia Tech, pela troca de conhecimentos e experiências. Em especial, gostaria de agradecer aos membros do LIS do IC Unicamp e do DLRL da Virginia Tech. Suas críticas, sugestões e comentários foram fundamentais para o aprimoramento deste trabalho. Agradeço, também, aos alunos de iniciação científica com quem venho trabalhando (Renata, Mauricio, Eduardo e Ricardo) por todo suporte; e a Marcos André Gonçalves e Dr. Hallerman, pela ajuda no desenvolvimento deste projeto na Virginia Tech.

A todos os funcionários da Unicamp e da Virginia Tech, cujo suporte e ajuda viabilizaram o desenvolvimento deste trabalho.

Este trabalho de pesquisa não seria possível sem o suporte financeiro da FAPESP (Procs. 00/04872-2 e 01/02788-7) e CAPES (Proc. BEX844/03-9), a quem dispenso meu sincero agradecimento.

Este trabalho foi também parcialmente financiado pelos projetos MCT-PRONEX SAI (Sisemas Avançados de Informação) e pelos projetos CNPq-WebMaps e Agroflow.

Sumário

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

1.1 Motivação

É crescente a preocupação com a conservação do meio ambiente. Para permitir iniciativas nesta área, é necessário disponibilizar um grande número de informações ambientais e softwares para gerenciá-las. Mudanças ambientais têm emergido como uma importante questão na agenda global. Conseqüentemente, há uma demanda por acesso seguro e atualizado das informações ambientais. Sistemas de informação ambientais são uma resposta a esta demanda. Eles visam o gerenciamento de dados sobre o meio ambiente, incluindo informações sobre o solo, a água, o ar e sobre as diversas espécies de animais e plantas existentes.

As aplicações ambientais têm como características marcantes o grande volume de dados envolvidos e o georreferenciamento destes dados (aplicações geográficas) [52]. Em alguns casos, os dados incluem também fotos de paisagens ou de seres vivos. No entanto, exceto no caso de imagens de sensoriamento remoto (satélite, radar, etc.), imagens são mantidas à parte do banco de dados propriamente dito, que fica restrito a dados espaciais. O ideal, em função da demanda dos usuários, seria integrar todos estes dados sob um único tipo de gerenciamento. Em especial, uma grande gama de aplicações ambientais requer consultas típicas em banco de dados de imagens – dentre outras, as chamadas *consultas por conteúdo* – mas não disponíveis nos sistemas existentes. Não existe, no entanto, ambiente que integre gerenciamento conjunto de fotos e de dados espaciais para aplicações ambientais.

As imagens de sensoriamento remoto são usadas em sistemas ambientais para determinar fatores como aspectos climáticos, pedológicos, estresse vegetal, poluição ou erosão. Tais dados são usados como base para estudo e simulação de ecossistemas e até mesmo influência do homem na natureza. Por outro lado, fotos obtidas em trabalho de campo (de seres vivos, ou de paisagens), apesar de serem uma importante fonte de informação,

não têm sido devidamente exploradas neste tipo de sistema.

O trabalho desenvolvido nesta tese contribui para resolver este problema. A tese aborda um tipo específico de sistema ambiental – aquele que trata de questões ligadas à biodiversidade. O trabalho realizado está centrado em combinar pesquisa em bancos de dados de imagens (e processamento de imagens por conteúdo) e aspectos de bancos de dados geográficos (e correlações espaciais). Esta combinação está baseada em um arcabouço de bibliotecas digitais, facilitando extensibilidade e reusabilidade. Como se verá no decorrer do texto, tal enfoque deu origem a um novo tipo de sistema para aplicações de biodiversidade.

1.2 Aspectos de Pesquisa Envolvidos

Os principais desafios a serem considerados em sistemas de informação ambientais são a necessidade de mecanismos de *interação* com usuário para facilitar a especificação de consultas, a dificuldade em combinar *mecanismos de consulta* por conteúdo a bancos de imagens e consulta a bancos de dados geográficos, e a complexidade do *gerenciamento* diferenciado de dados de natureza tão distinta. Estes desafios envolvem trabalhos em duas frentes: bancos de dados (contendo imagens e dados geográficos) e processamento de imagens. Além disso, como a solução adotada envolve bibliotecas digitais, o trabalho também precisou levar em consideração este fator.

Uma biblioteca digital pode ser vista como um sistema de informação complexo que provê uma coleção de recursos organizados, mecanismos para visualização e busca, ambientes distribuídos em rede e um conjunto de serviços, objetivando a satisfação das necessidades de usuários [93]. A biblioteca digital considerada nesta pesquisa disponibiliza coleções de dados sobre imagens de seres vivos, fenômenos geográficos e ecológicos, assim como mapas e metadados.

A pesquisa em bibliotecas digitais envolve descobertas em diferentes áreas tais como hiper-texto, recuperação de informação, serviços multimídia, gerenciamento de bancos de dados e interfaces [76]. O processo de construção de uma biblioteca digital envolve a especificação do conteúdo a ser armazenado, como estes conteúdos são organizados, estruturados, descritos e acessados, quais serviços são oferecidos pela biblioteca (visualização, busca, recomendação, etc.), e como usuários podem interagir com cada serviço oferecido pela biblioteca digital [76].

Há várias iniciativas na área de bibliotecas digitais que cobrem tópicos relacionados com a pesquisa da tese. Hong *et. al* [83], por exemplo, apresentam uma biblioteca digital de borboletas de Taiwan. Esta biblioteca armazena informações sobre o ecossistema das espécies e provê mecanismos para recuperação por conteúdo de imagens. Consultas baseadas na localização espacial das borboletas não são possíveis. Um outro exemplo

são as bibliotecas digitais de flora (FDL) [132], que compreendem repositórios contendo informações sobre taxonomia, mapas, ilustrações e descrições morfológicas de plantas. A recuperação de imagens por conteúdo não é prevista. Outras iniciativas referem-se a coleções de imagens de satélites [165] e de vídeos [29]. Nos dois casos, as bibliotecas digitais suportam a atribuição de referências espaciais (localização) aos documentos. No entanto, consultas que envolvam a combinação de predicados espaciais e recuperação baseada no conteúdo das imagens ou dos vídeos não são permitidas.

A contribuição de processamento de imagens em sistemas de biodiversidade envolve vários fatores. O principal consiste na especificação e definição de algoritmos que manipulam o conteúdo das imagens (objetos e suas propriedades de forma, cor e textura). Estes algoritmos visam extrair e descrever o conteúdo das imagens de forma que a descrição possa ser utilizada para indexar as imagens e manipulá-las segundo este conteúdo, em um banco de imagens. Para efeito da tese, o conteúdo de uma imagem é definido como o conjunto de objetos (flores, peixes, etc.) que a compõem. Embora vários projetos referenciem recuperação por conteúdo, a maioria destes usa características de toda imagem e não de seus objetos.

Um descritor de imagem pode ser caracterizado por: (i) um *algoritmo de extração de vetores de características* e (ii) uma *métrica de comparação* destes vetores. O *algoritmo de extração de vetores de características* é responsável por manipular o conteúdo da imagem (os objetos e suas propriedades tais como cor, textura e forma). Esta manipulação resulta na extração de uma série de *vetores de características*, que são conjuntos de valores numéricos que sumarizam o conteúdo de uma imagem. Por exemplo, as cores podem ser descritas usando histogramas [19, 148] que registram o número de *pixels* para uma dada cor. O conteúdo do descritor de cor seria assim um vetor de valores descrevendo o número de *pixels* de uma imagem com uma mesma cor. A descrição de textura, por sua vez, envolve muitas alternativas, tais como a matriz de co-ocorrência [81] ou representações baseadas em transformadas *wavelets* [95]. A descrição da forma é usualmente baseada no pré-processamento de imagens para extração dos seus objetos – uma técnica chamada segmentação [79] – e pela conseguinte caracterização de suas formas [37, 100, 162]. Exemplos de descritores de forma incluem *Curvature Scale Space* [1, 109] e *Beam Angle Statistics (BAS)* [6, 7].

Estes vetores de características podem ser usados, em seguida, para indexar as imagens e para manipulá-las segundo conteúdo em um banco de dados de imagens. Neste contexto, *métricas de comparação* são utilizadas para computar a distância entre dois vetores, e, portanto, a distância entre duas imagens. Ou seja, dois vetores de características de imagens são próximos se as imagens são similares; caso contrário, são distantes. Exemplos de métricas de distância incluem a distância Euclideana [53] ou algoritmos de casamento (*matching*) como o *optimal correspondent subsequence (OCS)* [156] usado na comparação

do descritor de forma *BAS* [6, 7].

Bancos de Dados de Imagens apresentam diversos desafios do ponto de vista de pesquisa e implementação, envolvendo problemas que vão desde questões de armazenamento até interfaces amigáveis [53, 129, 139]. O aspecto complicador reside no fato de que os objetos envolvidos (imagens) são muito mais complexos de gerenciar do que objetos textuais. Do ponto de vista de armazenamento, imagens ocupam muito espaço. Além disso, a indexação de imagens deixa de ser uma questão de processamento de *strings* e passa a depender de outras características, inclusive de diferentes aspectos cognitivos relativos à interpretação visual. Vários outros problemas - linguagens de consulta, atualização - contribuem para atrair cada vez mais pesquisadores para esta área. Vários exemplos de pesquisa em Bancos de Dados de Imagens são descritos na literatura [24, 71, 121, 122, 140, 153]. Estes sistemas são monolíticos e utilizam um conjunto de descritores pré-definido, o que dificulta a sua utilização em diferentes domínios.

Bancos de Dados Geográficos são repositórios de informação coletada empiricamente sobre fenômenos do mundo real (por exemplo, florestas, rios, cidades). À semelhança de bancos de dados de imagens, apresentam desafios tanto teóricos quanto de implementação. Parte destes desafios é devido à natureza intrínseca dos dados geográficos. Estes dados ocupam muito espaço e variam com o tempo. Além disso, são geralmente provenientes de fontes diferentes com níveis distintos de generalização e escalas incompatíveis. Outro aspecto complicador reside no fato de que a dimensão espacial introduz questões de restrição de integridade espacial e processamento de consultas espaciais [127]. E mais, estratégias padrão de otimização de consultas nem sempre são adequadas para dados geográficos [68]. O processamento de consultas em bancos de dados geográficos envolve três tipos de consulta – aquelas que retornam características de localização (para determinar "onde" determinado fenômeno ocorreu), aquelas relacionadas com características descritivas (para determinar "o que" foi encontrado em uma dada localização), e aquelas que retornam características temporais (para determinar "quando" um fenômeno ocorreu). Consultas mais complexas — tais como análise de tendência — podem ser especificadas como combinação destas três. As consultas espaciais podem, ainda, ser classificadas de acordo com os tipos de operador utilizados. Uma classificação comum distingue as consultas entre topológicas, métricas e direcionais [80].

A combinação de consulta por conteúdo e consulta espacial abre novas perspectivas para pesquisadores em biodiversidade. Um exemplo de consulta espacial em um sistema de biodiversidade é "Mostre as áreas onde a espécie de peixe *Percina Rex* foi observada". Desenhos e fotos de espécies também podem ser usados neste contexto. Eles são armazenados em arquivos de dados do sistema e são tratados como documentação auxiliar, sendo usualmente recuperados pelo nome da espécie. Já um exemplo de consulta envolvendo arquivos de imagens seria "Mostre todas as fotos contendo espécies do peixe *Percina*

Rex". Em outras palavras, as imagens são acessadas somente via consultas textuais, não havendo possibilidade de recuperação de imagem por conteúdo.

Idealmente, os pesquisadores em biodiversidade gostariam de combinar processamento de imagens de espécies com processamento de predicados textuais e espaciais. Um exemplo de consulta de imagem por conteúdo no contexto de biodiversidade utilizaria uma imagem de consulta (por exemplo, uma foto de um peixe) e pediria ao sistema para "Recuperar todas as imagens do banco de dados que contenham peixes que têm forma similar àquela apresentada na imagem de consulta". A combinação destas consultas consiste em "Mostre as áreas onde peixes da espécie *Percina Rex* coexistem com peixes cujas nadadeiras têm forma similar àquela apresentada na imagem de consulta".

Não se conhece um sistema de informação de biodiversidade que combine recuperação de imagem por conteúdo com gerenciamento de dados espaciais e convencionais. Esta tese contribui para solucionar este tipo de problema.

1.3 Objetivos e Contribuições

O objetivo da tese é contribuir para resolver problemas de especificação e implementação de sistemas de informação de biodiversidade que combinem o gerenciamento de imagens de seres vivos, descrições textuais e dados espaciais no contexto de bibliotecas digitais.

O trabalho utilizou dois conceitos básicos para especificação e implementação da arquitetura: (i) a noção de componentes de software, para facilitar reuso e extensibilidade; e (ii) o arcabouço de bibliotecas digitais, para permitir acoplamento de grandes coleções de dados de biodiversidade e seu gerenciamento. O uso destes dois conceitos permite a criação de sistemas sob perspectiva diferente do que é encontrado na maioria dos casos neste domínio. Os sistemas de informação de biodiversidade são em geral monolíticos e dedicados a tipos específicos de dados, reduzindo bastante seu uso genérico.

A figura 1.1, retirada do Capítulo 2, apresenta a arquitetura proposta para a criação de sistemas de informação de biodiversidade que gerenciem dados espaciais, dados textuais e imagens. Esta arquitetura é composta por três camadas: *BIS Manager*, responsável por receber consultas, processá-las, enviá-las para os componentes de busca apropriados, e por combinar os resultados de cada componente, enviando a resposta para o usuário final; por uma camada de componentes de busca, sendo uma para cada tipo de evidência; e os repositórios de dados.

As principais contribuições desta tese são:

1. Especificação de um ambiente para gerenciamento integrado de fotos, mapas e dados convencionais, que permite a cientistas em biodiversidade a extração de informação segundo suas necessidades [47, 49]. Este ambiente é ilustrado na Figura 1.1. A

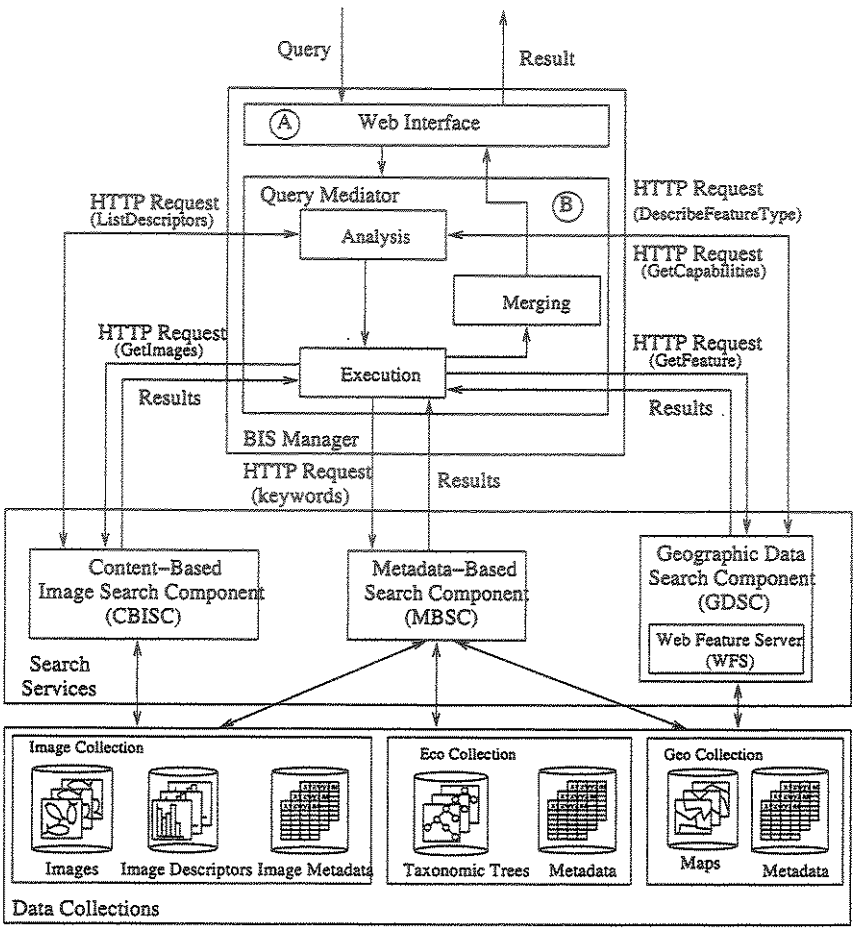


Figura 1.1: Arquitetura proposta para construção de Sistemas de Informação de Biodiversidade.

proposta deste ambiente é uma contribuição para desenvolvedores de sistemas de informação de biodiversidade.

2. Criação de um componente de busca para recuperação de imagens por conteúdo [48], que pode acomodar diferentes tipos de descritores e métricas de comparação. Esta é uma contribuição para a área de bibliotecas digitais, na medida em que desenvolvedores desta área podem usar este componente para suportar consultas de imagens por conteúdo. Esta contribuição está relacionada ao módulo C marcado na Figura 1.1.

3. Especificação e implementação de novos descritores de imagem baseados na forma de objetos [41, 42, 44, 50], validados através de testes experimentais. Experimentos

incluíram a avaliação de dezenas de descritores e o uso de diferentes coleções de imagens; apenas os melhores resultados são apresentados nesta tese. A proposta destes descritores representa contribuição para a área de processamento de imagens. Esta contribuição está relacionada ao módulo D marcado na Figura 1.1.

4. Proposta e desenvolvimento de novas estruturas visuais, visando novos tipos de interação em interfaces, para sistemas de recuperação de imagens por conteúdo [40]. Esta é uma contribuição para a área de interfaces, estando relacionada ao módulo A marcado na Figura 1.1.

5. Implementação parcial do ambiente, usando as demais contribuições, validando-o para um tipo específico de imagem (peixes) e perfil de usuários (ictiólogos), para suporte ao processo de identificação de espécies [47, 49]. Esta implementação parcial constitui contribuição para usuários de sistemas de biodoversidade, mais especificamente, para estudantes de biologia, professores e ictiólogos.

As contribuições, desta forma, cobrem não apenas a especificação do ambiente e sua validação, mas também aspectos em processamento de imagens, consultas que combinam parâmetros de várias naturezas e aspectos de interface humano-computador. Seus usuários passam a contar com uma variada combinação de operações até agora não disponíveis em sistemas existentes.

1.4 Organização da Tese

O texto desta tese está organizado agrupando os principais artigos publicados e/ou submetidos para publicação que foram resultado da pesquisa realizada. Pequenas correções foram feitas no texto original destes artigos (definições, notações, erros de ortografia, etc.), de modo a manter a consistência do texto final da tese. Os Capítulos 2 e 3 tratam de contribuições de combinação de processamento de imagens, consultas em bancos de dados espaciais e bibliotecas digitais; os Capítulos 4 e 5 apresentam resultados em processamento de imagens; o Capítulo 6 apresenta resultados em interfaces e o Capítulo 7 conclui a tese.

1.4.1 Capítulo 2

O Capítulo 2 (*Integrating Image and Spatial Data for Biodiversity Information Management Systems*) inclui os resultados apresentados em [49].

Um exemplo de uma consulta espacial padrão em sistemas de biodiversidade seria "Mostre todas as áreas onde a espécie de peixe *Percina rex* tem sido observada". Uma

consulta típica de correlação espacial requer a combinação de informação sobre a localização de espécies e condições climáticas. Desenhos e fotos de espécies também podem ser usados neste contexto. Eles são armazenados à parte em arquivos de dados, sendo tratados como documentação auxiliar, usualmente sendo recuperados a partir dos nomes das espécies. Um exemplo de uma consulta envolvendo arquivos de imagens seria "Mostre todas as fotos do peixe *Percina rex*". Imagens são acessadas apenas a partir de consultas textuais, ignorando-se recuperação de imagem por conteúdo.

O gerenciamento de imagens baseado em conteúdo, por outro lado, permite a cientistas identificar espécies usando uma dada imagem (i.e., uma foto) e buscar em um banco de dados por imagens "mais similares". A distribuição geográfica, neste caso, é armazenada como metadado textual (i.e., nomes de regiões), não sendo possível realizar correlações espaciais.

O objetivo deste trabalho é a criação de Sistemas de Informação de Biodiversidade que combinam estes tipos de características de busca através de consultas exploratórias. Este SIB ajudará cientistas a melhorar ou completar seu conhecimento e entendimento sobre espécies e seus habitats ao combinar consultas textuais, consultas baseadas no conteúdo de imagem e consultas geográficas. Um exemplo de consulta deste tipo poderia começar por definir uma imagem de entrada (por exemplo, uma foto de peixe), e então pedir ao sistema que "Recupere todas imagens do banco de dados contendo peixes cujas nadadeiras têm forma similar àquelas do peixe mostrado na foto". Uma combinação desta consulta com predicados textuais e espaciais consistiria em "Mostre as bacias hidrográficas onde as espécies de peixe com 'olhos grandes' coexistem com peixes cujas nadadeiras são similares às do peixe da foto".

Desafios incluem trabalhos em duas frentes: processamento de imagens e bancos de dados. Sistemas disponíveis não atacam estas questões simultaneamente — eles se concentram em dados de imagem ou dados espaciais. De fato, SIGs que suportam consultas de imagens lidam com correlações espacias e não com características de imagens tais como cor ou textura. Nosso trabalho, ao contrário, combina estas fontes de evidência beneficiando-se de facilidades de bibliotecas digitais, que oferecem uma infra-estrutura organizada para integrar rede de coleções de dados heterogêneos. Estes dados incluem imagens de seres vivos e distribuição geográfica, assim como mapas e metadados ecológicos, geográficos e de imagens. Nossa solução está sendo instanciada em um Sistema de Informação de Biodiversidade para espécies de peixe em uma aplicação real. O objetivo é ajudar estudantes, pesquisadores e membros do público em geral a identificar espécies de peixes usando as ferramentas de consulta disponíveis na arquitetura proposta.

1.4.2 Capítulo 3

O Capítulo 3 (*A Digital Library Framework for Biodiversity Information Systems*) inclui os resultados apresentados em [47].

O objetivo da pesquisa apresentada neste trabalho é combinar resultados obtidos em processamento de imagem, bancos de dados e bibliotecas digitais para prover pesquisadores em biologia com um sistema de informação de biodiversidade que integre consultas envolvendo tanto conteúdo de imagens quanto dados textuais. Neste contexto, por exemplo, usuários somente precisarão prover uma imagem de consulta (i.e., uma foto de uma folha de uma planta e solicitar ao sistema que "mostre todas as áreas do Brasil onde a espécie de planta *Acacia polyphylla* coexiste com plantas cujas folhas têm formato similar à apresentada na foto".

Com esta finalidade, este trabalho apresenta uma arquitetura de biblioteca digital genérica para gerenciamento de dados heterogêneos sobre seres vivos e seus ecossistemas. Estes dados envolvem não somente catacterísticas textuais e de localização, mas também imagens destes seres vivos. Uma noção chave considerada é a de *componente de bibliotecas digitais*, um módulo de software especialmente projetado para encapsular funcionalidade, daí suportando modularidade, flexibilidade e reuso na construção de infra-estrutura de bibliotecas digitais. Devido ao seu projeto baseado em componentes, esta arquitetura resolve o problema de interoperabilidade que sistemas existentes de informação de biodiversidade apresentam. Para ilustrar o uso desta arquitetura em uma aplicação real, ela foi instanciada para suportar a criação de um sistema de informação para espécies de peixes. O objetivo deste sistema é ajudar pesquisadores em Ictiologia no processo de identificação de espécies usando técnicas de recuperação disponíveis na arquitetura proposta.

As principais contribuições deste trabalho são as seguintes: (a) uma arquitetura genérica para gerenciamento de coleções heterogêneas, baseada em componentes de bibliotecas digitais, para acessar fontes de dados heterogêneos (texto e imagens), que permitem a combinação de consultas baseadas em texto com consultas baseadas no conteúdo de imagem; e (b) um novo componente, para busca de imagem por conteúdo, integrado à arquitetura proposta. Este componente, recentemente proposto em Torres et al [48], suporta o uso dos descritores de forma apresentados nos Capítulos 4 e 5.

1.4.3 Capítulo 4

O Capítulo 4 (*A Graph-based Approach for Multiscale Shape Analysis*) inclui os resultados apresentados em [44].

Este trabalho apresenta os resultados referentes à proposta de novos descritores de forma (dimensão fractal multi-escala do contorno — *Contour Multiscale Fractal Dimension* — e saliências do contorno — *Contour Saliences*), utilizados no suporte à recuperação

de imagens por conteúdo. O cálculo deste descritores usa a Transformada Imagem-Floresta (*image foresting transform* — IFT), uma abordagem baseada em grafo para projeto de operadores de processamento de imagens [63–65, 67, 101]. Neste caso, os descritores de forma são obtidos a partir de representações criadas pela IFT.

A dimensão fractal multi-escala [36, 39] é um conceito novo, que soluciona muitos dos problemas dos métodos existentes [110, 120] para estimação numérica da dimensão fractal. A dimensão fractal de uma forma é calculada através da Transformada de Distância Euclideana (TDE) dos seus pixels. A TDE destes pixels é também relacionada com seus diagramas de Voronoi geométrico [126], onde cada pixel define uma zona de influência (*região discreta de Voronoi*) composta pelos pixels mais próximos. As saliências de uma forma são calculadas a partir das áreas das regiões Voronoi dos pixels de mais alta curvatura dentro de uma região estreita ao redor da forma [36]. Esta abordagem permite a quantificação dos valores de curvatura (centro dos pixels) onde a curvatura analítica seria infinita. A IFT provê o cálculo simultâneo da TDE e das regiões discretas de Voronoi em tempo usualmente proporcional ao número de pixels [65], sendo mais eficiente do que o método proposto em [36].

Este trabalho também introduz melhoramentos no cálculo da dimensão fractal multi-escala e das saliências do contorno. A abordagem original para cálculo da dimensão fractal multi-escala sofre com oscilações indesejadas na curva fractal, e a localização dos pontos de maior curvatura ao longo do contorno, importante para cálculo das saliências, é muito sensível no caso de formas complexas e intricadas. O problema da oscilação é resolvido usando regressão polinomial. A relação entre pontos de saliência de um contorno e dos seus esqueletos interno e externo — um conceito importante introduzido por [97] — é usada para localizar os pontos de maior curvatura ao longo do contorno, melhorando consideravelmente a robustez do cáculo de suas saliências. Esta relação é obtida de maneira direta através do arcabouço da IFT. O descritor de saliência de contorno é também redefinido de modo a incluir a localização e o valor de saliência ao longo do contorno e uma métrica de distância especial, o que produz uma alta eficácia no reconhecimento de formas.

Os descritores propostos são comparados com a dimensão fractal simples, dois descritores clássicos (*Fourier descriptors* [154] e *moment invariants* [84]) e dois descritores recentemente publicados (*Curvature Scale Space (CSS)* [1, 109] e *Beam Angle Statistics (BAS)* [6, 7]) no que diz respeito aos seguintes aspectos: *compactabilidade* e *separabilidade*. A compactabilidade de um descritor indica sua invariância a variações de objetos pertencentes a uma mesma classe, enquanto a separabilidade indica sua habilidade de discriminar objetos que pertencem a classes diferentes. Em outras palavras, um descritor é considerado "bom" quando ele cria agrupamentos compactos bem separados uns dos outros, para todas as classes em um espaço de características correspondente.

1.4.4 Capítulo 5

O Capítulo 5 (*Contour Salience Descriptors for Effective Image Retrieval and Analysis*) inclui os resultados apresentados em [41].

Este trabalho apresenta os resultados referentes à proposta do descritor de forma chamado de saliências do segmento do contorno — *Contour Segment Saliences*, utilizado no suporte à recuperação de imagens por conteúdo. Além disso, ele apresenta uma nova estratégia para incorporar as saliências côncavas no descritor saliências do contorno, definido em [44] (Capítulo 4).

Costa *et al.* [36] propuseram o uso de *saliências de forma* para a representação de objetos. As saliências de uma forma são definidas como as áreas de influência máxima dos pontos de mais alta curvatura, considerando uma *região estreita* em ambos os lados da curva e as regiões de Voronoi dos seus pontos. Um ponto de contorno, por exemplo, é considerado convexo quando sua área de influência é maior fora do que dentro do contorno, e côncavo, caso contrário. Uma região estreita é usada para reduzir o quanto seja possível a influência cruzada de partes opostas de um contorno complexo. Torres *et al.* [42] apresentaram uma maneira mais eficiente para calcular as saliências de uma forma usando a Transformada Imagem-Floresta [67] e um descritor de saliência do contorno para recuperação [50] e análise [44] de imagem. Em ambos os trabalhos, as saliências do contorno foram comparadas com vários outros descritores, incluindo o *curvature scale space* [1, 109] e o recentemente proposto *beam angle statistics* [6, 7]. Entretanto, as saliências do contorno nunca consideraram os pontos de saliência côncava, porque sua efetividade era muito sensível à localização precisa destes pontos. Este trabalho resolve este problema incorporando os pontos côncavos ao descritor de saliência do contorno. Além disso, ele propõe um outro descritor baseado nos valores de saliência de segmentos do contorno. O contorno é dividido em um número fixo de segmentos e as áreas de influência dos seus pixels dentro e fora do contorno são usadas para calcular as *saliências do segmento*. O descritor de saliência de segmento consiste nos valores de saliência dos segmentos de contorno e de um algoritmo de casamento como função de distância. Este trabalho também discute novos resultados experimentais que mostram a superioridade deste descritor no que diz respeito às métricas separabilidade e compactabilidade.

1.4.5 Capítulo 6

O Capítulo 6 (*Visual Structures for Image Browsing*) inclui os resultados apresentados em [40].

Tipicamente, o resultado de uma consulta em um banco de dados de imagens é um conjunto de imagens, mostradas em um Visualizador. Infelizmente, estes conjuntos são usualmente extensos, o que dificulta o processo de visualização e/ou exploração de re-

sultado. A técnica de apresentação de resultado mais comum é baseada em se mostrar uma matriz de duas dimensões de versões em miniatura de imagens [71, 117]. Esta matriz é organizada de acordo com a similaridade de cada imagem retornada em relação ao padrão de consulta (i.e., da esquerda para direita, de cima para baixo). Trata-se de uma matriz $n \times m$, onde a posição $(1, 1)$ é ocupada pela miniatura do padrão de consulta, a posição $(1, 2)$ pela imagem mais similar a ele, e assim por diante. Este método facilita a visualização, permitindo aos usuários que eles percorram o conjunto de imagens como se estivessem lendo um texto [128]. Esta abordagem, entretanto, mostra as imagens com diferentes graus de similaridade com a mesma distância da imagem de consulta: i.e., imagens $(1, 2)$ e $(2, 1)$ são mostradas com a mesma distância física do padrão de consulta, mas a primeira é mais similar do que a última. Outras abordagens para visualização tentam considerar a similaridade relativa não somente entre o padrão de consulta e cada imagem recuperada, mas também entre todas as imagens retornadas [133, 143]. Estas iniciativas têm como desvantagem o fato de que imagens similares que são colocadas próximas umas das outras parecem se sobrepor, sendo menos atraentes do que se estivessem separadas [128].

Este trabalho apresenta uma nova abordagem para estes problemas de interação de usuários. Esta abordagem é baseada na adoção de técnicas de Visualização de Informação para prover usuários com apresentações de resultados semanticamente enriquecidos, e novos tipos de mecanismos de interação. Visualização de Informação é um importante campo dentro da Interação Homem-Computador (IHC) que objetiva o estudo e o uso de representações visuais interativas para abstração de dados de modo a ampliar cognição [22, 28, 137].

As principais contribuições deste trabalho são as seguintes:

- apresentação de duas técnicas de visualização baseadas em *Espiral* e *Anéis Concêntricos* para explorar resultados em bancos de dados de imagens. Estas técnicas possibilitam a usuários novos meios de ranqueamento de imagens similares sem sobreposições;

- descrição de um protótipo de sistema de recuperação de imagem por conteúdo que incorpora estes paradigmas de visualização. As propriedades de visualização e interação do protótipo são baseadas no modelo de referência descrito em [22].

Finalmente, o Capítulo 7 conclui a tese, sumarizando suas contribuições e extensões.

1.4.6 Outras Publicações

Além destes trabalhos principais que compõem o corpo da tese, os seguintes trabalhos foram publicados durante a pesquisa:

sultado. A técnica de apresentação de resultado mais comum é baseada em se mostrar uma matriz de duas dimensões de versões em miniatura de imagens [71, 117]. Esta matriz é organizada de acordo com a similaridade de cada imagem retornada em relação ao padrão de consulta (i.e., da esquerda para direita, de cima para baixo). Trata-se de uma matriz $n \times m$, onde a posição $(1, 1)$ é ocupada pela miniatura do padrão de consulta, a posição $(1, 2)$ pela imagem mais similar a ele, e assim por diante. Este método facilita a visualização, permitindo aos usuários que eles percorram o conjunto de imagens como se estivessem lendo um texto [128]. Esta abordagem, entretanto, mostra as imagens com diferentes graus de similaridade com a mesma distância da imagem de consulta: i.e., imagens $(1, 2)$ e $(2, 1)$ são mostradas com a mesma distância física do padrão de consulta, mas a primeira é mais similar do que a última. Outras abordagens para visualização tentam considerar a similaridade relativa não somente entre o padrão de consulta e cada imagem recuperada, mas também entre todas as imagens retornadas [133, 143]. Estas iniciativas têm como desvantagem o fato de que imagens similares que são colocadas próximas umas das outras parecem se sobrepor, sendo menos atraentes do que se estivessem separadas [128].

Este trabalho apresenta uma nova abordagem para estes problemas de interação de usuários. Esta abordagem é baseada na adoção de técnicas de Visualização de Informação para prover usuários com apresentações de resultados semanticamente enriquecidos, e novos tipos de mecanismos de interação. Visualização de Informação é um importante campo dentro da Interação Homem-Computador (IHC) que objetiva o estudo e o uso de representações visuais interativas para abstração de dados de modo a ampliar cognição [22, 28, 137].

As principais contribuições deste trabalho são as seguintes:

- apresentação de duas técnicas de visualização baseadas em *Espiral* e *Anéis Concêntricos* para explorar resultados em bancos de dados de imagens. Estas técnicas possibilitam a usuários novos meios de ranqueamento de imagens similares sem sobreposições;

- descrição de um protótipo de sistema de recuperação de imagem por conteúdo que incorpora estes paradigmas de visualização. As propriedades de visualização e interação do protótipo são baseadas no modelo de referência descrito em [22].

Finalmente, o Capítulo 7 conclui a tese, sumarizando suas contribuições e extensões.

1.4.6 Outras Publicações

Além destes trabalhos principais que compõem o corpo da tese, os seguintes trabalhos foram publicados durante a pesquisa:

- *Shape Description by Image Foresting Transform* [42]: Este artigo introduz os descritores de forma *Contour Multiscale Fractal Dimension* e *Contour Saliences*, usando o arcabouço da Transformada Imagem-Floresta (Image Foresting Transform – IFT). A IFT é uma nova metodologia para a redução de um problema em processamento de imagens em um problema de floresta de caminho de custo mínimo em um grafo, cuja solução pode ser obtida em tempo linear [67].

- *Effective Image Retrieval by Shape Saliences* [50]: Este artigo apresenta mais detalhes relativos à extração do descritor de forma *Contour Saliences*, inicialmente apresentado em [44]. Além disso, ele descreve resultados experimentais usando métricas como precisão e revocação com intuito de comparar o descritor proposto com outros descritores clássicos da literatura.

- *Using Digital Library Components for Biodiversity Systems* [46], pôster apresentado no *ACM-IEEE Joint Conference on Digital Libraries (JCDL 2004)*. Ele introduz o uso de componentes de Bibliotecas Digitais na construção de sistemas de informação de biodiversidade (veja Capítulos 2 e 3).

- *An OAI Compliant Content-Based Image Search Component* [48], relacionado a uma demonstração realizada no *ACM-IEEE Joint Conference on Digital Libraries (JCDL 2004)*. Esta demonstração teve como objetivo apresentar um novo componente de busca de imagens por conteúdo. Este componente foi utilizado na implementação da arquitetura proposta para sistemas de informação de biodiversidade (veja Capítulo 3).

Capítulo 2

Integrating Image and Spatial Data for Biodiversity Information Management

2.1 Introduction

Biodiversity Information Systems – BIS (e.g., [5, 15, 16]) – involve huge sets of geographic data as well as large databases concerning species' descriptions (e.g., taxonomic classifications). Most biodiversity systems are concerned with determining spatial distribution of one or more living species, and the spatio-temporal correlations and trends of these distributions. This requires combining data on species (when and where they are observed, by whom and how) with geographic data (characterizing the ecosystems where the species are observed). Data integration usually is based on spatial properties, and thus Geographic Information Systems (GIS) and geographic databases are essential to develop this kind of system.

An example of a standard spatial query in a biodiversity system is "Show the areas where the fish species *Percina rex* has been observed". A typical spatial correlation query requires combining information on species location and climatic conditions. Drawings and photos of species also may be used in this context. They are stored apart in data files, and treated as auxiliary documentation, usually being retrieved by species' name. One example of a query involving image files would be "Show all photos of fish species *Percina rex*". Images are accessed only via textual queries, ignoring content-based image retrieval. In these systems, scientists must always search for specific species by name.

Content-based image management, on the other hand, allows scientists to identify species using a given image (e.g., a photo) and search in a database for the "most similar" images. Geographic distribution, in this case, is stored as textual metadata (e.g., names

of regions), and spatial correlations are infeasible.

The objectives here is to provide biodiversity researchers with a BIS that combines these types of searching characteristics for exploratory querying. This BIS will help scientists to enhance or complete their knowledge and understanding about species and their habitats by combining textual, image content-based and geographical queries. An example of such a query might start by providing an image as input (e.g., a photo of a fish) and then asking the system to "Retrieve all database images containing fish whose fins are shaped like those of the fish in this photo". A combination of this query with textual and spatial predicates would consist of "Show the drainages where the fish species with "large eyes" coexists with fish whose fins are shaped like those of the fish in the photo".

Challenges involve work on two fronts: image processing and databases. Available systems do not attack these questions simultaneously – they either concentrate on image data or on spatial data. Indeed, GIS that support image queries are concerned with spatial correlations and not with image features (such as color or texture features). Our work, instead, combines these sources of evidence taking advantage of digital library facilities, which offer an organized infrastructure to integrate networked collections of heterogeneous data. These data consist of images of the living beings and geographic distribution, as well as maps and geographic, ecological, and image metadata. Our solution is being instantiated in a BIS for fish species in a real application. The goal is to help students, researchers, managers, and members of the general public to identify fish specimen by using retrieval techniques.

This text is organized as follows. Section 2.2 outlines the architecture of the BIS. Section 2.3 discusses the application scenario that instantiates the architecture for an ichthyology biodiversity system. Section 2.4 gives a brief introduction to related research. Finally, Section 2.5 presents conclusions and ongoing work.

2.2 Architecture

Figure 2.1 shows the basic architecture proposed for biodiversity information management. This architecture is composed by three main layers: data collections (see Section 2.2.1), search services (Section 2.2.2), and BIS Manager (Section 2.2.3).

Collections are organized in a digital library comprised of a set of search services. The BIS combines textual queries with image processing algorithms to extract image descriptors, and spatial data management in geographic databases based on location and on ecological features.

Although this architecture has been specified in a generic way, its implementation is being carried out for particular fish species. Thus, image data consists of fish photos,

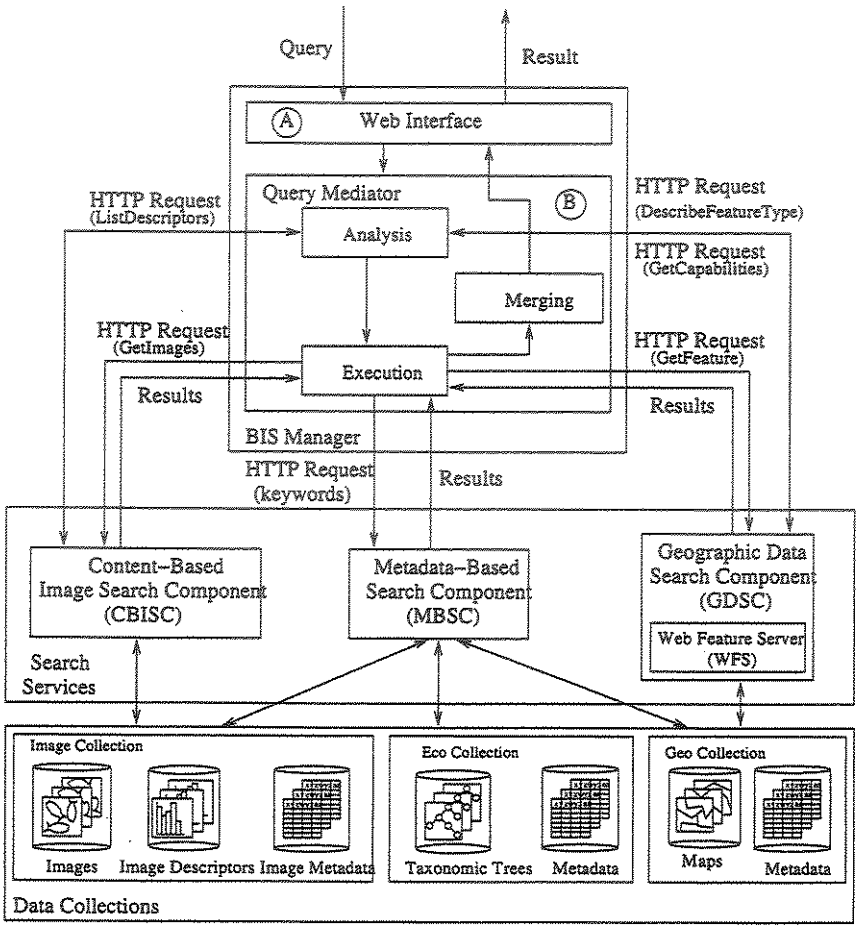


Figura 2.1: BIS System Architecture.

geographic data concerning areas where these fishes are likely to be found, and biodiversity metadata on fish and their ecosystems. As will be seen in Section 2.3, a considerable part of this architecture has already been implemented and tested. We are now working on the final integration with the geographic data search component.

2.2.1 Data Collections

This layer is responsible for database storage and low-level data management – image, geographic, and metadata databases. Sections 2.3.2 and 2.3.3 comment on the data collections used in the present implementation.

2.2.2 Search Services

Three search components are provided: a geographic data search component, a content-based image search component called *CBISC*, and a metadata-based search component called *ESSEX*.

Geographical Data Search Component

The *Geographical Data Search Component (GDSC)* encapsulates a Web Feature Server (WFS) [158], an OpenGIS consortium [118] recommendation for fostering interoperability. It defines an interface allowing requests for geographical features across the Web, and uses the XML-based Geography Markup Language (GML) [75] for data exchange. GML utilizes XML to express geographical features. It can serve as a modeling language for geographic systems as well as an open interchange format for geographic data. We are using the GeoServer [74] free implementation of OpenGIS Consortium's WFS implementation specification.

The WFS submodule of the *GDSC* is responsible for performing queries on the data sources: it receives HTTP requests from the client (the *Execution* submodule in Figure 2.1) and returns results as a GML or XML document, depending on the request. A WFS request consists of a description of a query or a data transformation operation, applied to one or more features. Available operations include:

GetCapabilities: A WFS must be able to describe its capabilities. Specifically, it must indicate which feature types and what operations are supported on each feature type. For example, it could define that a *feature type* named *ekey:fishspecies* encoding occurrences of fish species within a specific region is available. It also could indicate supported operations on this feature type (such as operations based on spatial predicates – e.g., *Intersect*, *Within*, etc.).

DescribeFeatureType: A WFS must be able, upon request, to describe the structure of any feature it can service.

GetFeature: A WFS must be able to answer a request, and retrieve feature instances.

Content-Based Image Search Component

One of the most common approaches to image retrieval is based on the so-called *Content-Based Image Retrieval (CBIR)* systems. Basically, these systems try to retrieve images similar to a user-defined pattern (e.g., image example). Their goal is to support image retrieval based on *content* properties (e.g., shape, color, or texture), which are often encoded in terms of *image descriptors*.

The *Content-Based Image Search Component (CBISC)* is a new search engine recently developed to support content-based queries on image collections [48]. It supports retrieval

using color, shape, and texture descriptors, with the corresponding feature vectors, stored in 1D or 2D structures. *CBISC* encapsulates metric index structures [31] to speed up the search process.

CBISC is based on the Open Archives Initiative (OAI) [116, 145] principles. The OAI develops and promotes interoperability standards that aim to facilitate the efficient dissemination of content.

Our *CBISC component* is an OAI-like search component that supports queries on image content. As in the OAI protocol [94, 116], queries are given by way of HTTP requests. However, we generalize to have an extended OAI (XOAI) protocol for image search that fits into the Open Digital Library (ODL) framework [145, 147]. As is typical with XOAI protocols, each request specifies the Internet host of the HTTP server and gives a list of key-value pairs. Two different requests ("verbs") are supported by this image search component:

ListDescriptors: Used to retrieve the list of image descriptors supported. No arguments are required for this verb.

GetImages: Used to retrieve a set of images by taking into account their contents. Required arguments specify the query image, the descriptor to be used, and the kind of query. The present version of *CBISC* supports two kinds of queries: *K-nearest neighbor query (KNNQ)* and *range query (RQ)* [31].

Metadata-Based Search Component (MBSC)

The *ESSEX* search engine [59] is being used as our metadata-based search component. *ESSEX* is a componentized vector-space search engine optimized for digital libraries. *ESSEX* acts as the core portion of an Open Digital Library (ODL [145]) search component, answering requests transmitted through an extended OAI (XOAI) protocol. *ESSEX*, available as open source software, was primarily developed for the CITIDEL (Computing and Information Technology Interactive Digital Educational Library) project [32], and also is being used in the *PlanetMath* project [123]. In *ESSEX*, all information is indexed in "chunks" associated with field names, where chunks may correspond to XML elements in a metadata record. Its high speed is the result of both keeping index structures in memory and using a background daemon model based on socket communication with the DL application.

2.2.3 BIS Manager

This module comprises a Web interface and a query mediator.

A) Web Interface:

This interface supplies query specification and visualization of results. The user will be able to formulate textual queries, interactive queries on maps, queries for image content, or a combination of these.

B) Query Mediator:

The search services are supported by a *Query Mediator* implemented as a server, which combines query mechanisms in metadata, image, and geographic data collections.

Its *Analysis* submodule receives as input a specification in terms of a query image, query terms, and/or rectangle coordinates in a map – and parses it. The parsing process takes advantage of previous knowledge of the *GDSC* and *CBISC*. In the former case, this information is obtained in the form of XML and XML schema documents, obtained each time it performs *GetCapabilities* and *DescribeFeatureType* requests on the *GDSC*. In the latter case, the *Analysis* submodule performs a *ListDescriptor* request on *CBISC* to obtain the enumeration of the image descriptors supported. Note that this information also can be used to guide the query optimization process.

The *Execution* submodule is in charge of decomposing the original query into subqueries and forwarding them to the appropriate search component (*CBISC*, *MBSC*, or *GDSC*). Finally, the *Merging* submodule combines the obtained results by using an appropriate combination scheme, and returns a ranked list containing the "most" similar objects matching the original specification.

2.3 Application Scenario and Implementation

The application scenario concerns the instantiation of the proposed architecture to support the creation of a BIS for fish species in a real application. The goal is to help fisheries students, researchers, managers, and the general public to identify fish specimens by using search retrieval techniques. This system will be used by students in ichthyology courses of the Department of Fisheries and Sciences at Virginia Tech.

2.3.1 Problem

Given a mixed collection of specimens from a river, ichthyologists face the problem of identifying which fish species are present in that collection. Their aim is to determine the taxonomic classification (e.g., family, genus, species) of each given specimen. The traditional approach is based on the use of dichotomous *keys* – basically, rules defining a decision tree that is traversed until one reaches an identification (e.g., [89]).

Operationally, this approach suffers from several problems. First, while an experienced scientist knows how to answer technical questions on subtle features of fish anatomy in order to use a dichotomous key, a student or non-scientist will find it difficult or impos-

sible to correctly answer those questions. Second, dichotomous keys often lack images to support their use by non-experts. Third, dichotomous keys invariably lack reference to geographic distributions of fishes, although geographic data can prove highly useful for fish identification. For example, knowing only where they collected a specimen, novices often have difficulty making species identifications. Access to geographically explicit information on fishes occurring in a watershed – especially if related to information on shapes and other appearance-related characters of the respective species – can aid in fish identification, reliably to genus and often to species. Often, a worker has a preliminary idea of the genus and species of a specimen. Knowing where the fish was collected, identification of the specimen is facilitated by access to information on the particular species occurring in that watershed. Certain families of freshwater fishes, for example, the sculpins (Family *Cottidae*), contain a number of cryptic species that are difficult to differentiate. Species identification is supported by taking into account spatial correlations among fish observations.

Our BIS tries to solve these problems, starting from the key-based approach, by creating a fish identification system that instead of being merely based on textual definitions, improves the fish identification process by allowing users to perform successive queries based on fish shape information, textual descriptions, and geographical data.

2.3.2 Present Implementation Stage

The *MBSC* and *CBISC* architecture components are fully operational for the goal application, being already tested through a Web-based application interface [45]. In particular, we have tested different kinds of queries on a collection of 11000 fish images [40]. The present version is to be used for fish identification in the Commonwealth of Virginia, and thus is restricted to fishes found in this area.

Our data collections comprise an image database (with fish photos); a geographic database (containing spatial data characterizing the regions in which the fish have been observed); and a database with metadata on the fish species and on the geographic data, dichotomous keys for identifying fishes and fish taxonomic trees. Metadata help query processing and are stored in a PostgreSQL database, while the current *CBISC* version manages image content description as XML documents.

We are now working on both implementing the BIS Manager modules with respect to the geographic data handling and organizing the Geo collection database. The latter uses PostgreSQL [125] database system and PostGIS [124]. PostGIS can be seen as an OpenGIS-conformant extension to the PostgreSQL, which allows geographic information systems objects to be stored into the database.

2.3.3 Data Sources

The fish-related data were obtained from [89] and from a site recently created to help students in the fish identification process [82]. Fish keys and metadata include data about over 200 species found in the Commonwealth of Virginia, USA. A subset of these data, covering 183 species and 187 images, is being used in this work.

Biodiversity Metadata: The biodiversity-related metadata include data about fish taxonomic classification (species, genus, family), common names, reproductive and food habits, metabolism, habitat description, information about similar species, and morphological descriptions.

Image Description: Current experiments configured *CBISC* to use several shape descriptors [7, 44]. We will further extend it to support queries on color information [148].

Geographic Data: The geographic data include maps (encoded in the ShapeFile format), spatial, and conventional data characterizing the regions in which the fish have been observed. Coordinates referring to the locations of occurrence of fish species also are stored. Data are being obtained from the Conservation Management Institute (CMI) at Virginia Tech. The CMI's Fish and Wildlife Information Exchange (FWIE) Division works as a technical assistance center, data analysis center, and information clearinghouse for fish, wildlife, and land management agencies and organizations.

2.3.4 Identifying a Specimen

An example of a query including textual, geographic data and image descriptor information is: *"retrieve fish descriptions of all fish whose shape is similar to that shown in Figure 2.2, which belong to genus 'Notropis', which have 'large eyes' and 'dorsal stripe', and have been observed within the catchments of the 'Tennessee' river"*. Notice that the first part of this query (shape similarity) is typical of image information systems; genus and physical characteristics are extracted from metadata-based systems; the last part is typical of GIS-based biodiversity systems (using the "within" spatial operation). The geographic component of the query is typically processed using a buffer operator or a user-specified rectangle encompassing part of the Tennessee drainage.

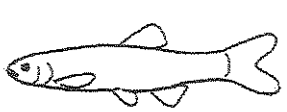


Figura 2.2: Example of shape outline used to define a query.

This query requires processing data from a variety of heterogeneous sources, stored in different formats. These sources are composed of images, image metadata and content de-

scriptors, ecology-related data (species description and taxonomic trees), and geographical information (spatial data and metadata).

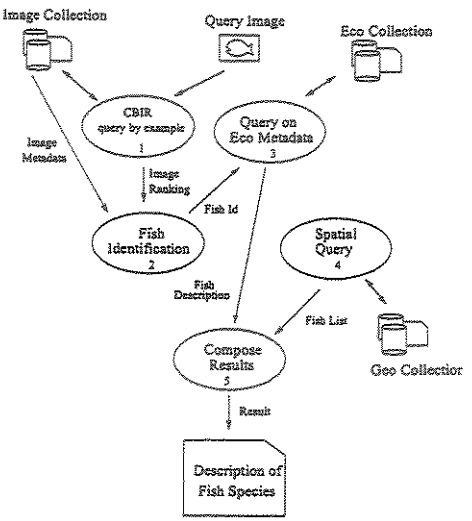


Figura 2.3: Execution plan for identifying fish species.

Figure 2.3 illustrates a possible execution plan for the proposed query within our BIS. It is composed of several steps, represented by ellipses. First, a content-based image retrieval process is executed in an image collection. Here, the *Execution* submodule of the *Query Mediator* (see Figure 2.1) performs a *GetImage* HTTP request on *CBISC*, using the image showed in Figure 2.2 as input and a pre-defined shape descriptor (1). CBISC will return a list of images, ranked by similarity to the input image.

The list of similar images is next used to retrieve fish identification parameters (2) for each image. Next, a query is performed on the *MBSC* to return fish species that belongs to genus "Notropis", whose morphological description include terms like "large eyes" and "dorsal stripe", and whose identification parameters match those returned by *CBISC* (3).

In the following, a spatial query is executed in order to identify which fishes have been observed within the catchments of the "Tennessee river" (4). This query is performed using a *GetFeature* HTTP request on the *GDSC*. By considering the rectangle-based query, this HTTP request might be encoded in XML as shown in Figure 2.4, where the query Filter is a box. The result of the query is a list of fish species' names and scientific names (parameters Property Name).

Finally, the results of (3) and (4) are combined by the *Merging submodule* and the descriptions of the most relevant fish species are returned (5).

This is, of course, one possible processing strategy. Another alternative would be to start with a spatial query that would limit the set of fish species to those observed within

```
<wfs:GetFeature service="WFS" version="1.0.0"
 outputFormat="GML2"
 xmlns:topp="http://www.openplans.org/topp"
 xmlns:wfs="http://www.opengis.net/wfs"
 xmlns:ogc="http://www.opengis.net/ogc"
 xmlns:gml="http://www.opengis.net/gml"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/wfs
                http://schemas.opengis.net/wfs/1.0.0/WFS-basic.xsd">
 <wfs:Query typeName="ekey:fishspecies">
  <wfs:PropertyName>ekey:scientific_name</wfs:PropertyName>
  <wfs:PropertyName>ekey:common_name</wfs:PropertyName>
  <ogc:Filter>
   <ogc:BBOX>
    <ogc:PropertyName>the_geom</ogc:PropertyName>
    <gml:Box srsName="http://www.opengis.net/gml/srs/epsg.xml#27345">
     <gml:coordinates>489154,5433017 505234,5448023</gml:coordinates>
    </gml:Box>
   </ogc:BBOX>
  </ogc:Filter>
 </wfs:Query>
</wfs:GetFeature>
```

Figura 2.4: Example of WFS XML request which fetches fish species (feature) with a bounding box filter.

a certain range of the catchments. Next, content-based retrieval would be applied only to those species.

The existence of alternatives to query processing concern another issue, that of optimization. Our work is not yet concerned with performance aspects, and so assumes a predefined query processing strategy.

2.4 Related Work

The work proposed here involves combining research on image databases, geographic databases, and digital libraries for biodiversity information management. The following paragraphs outline related work in these areas.

Biodiversity Information Systems and GIS: There are several initiatives for the development of biodiversity information systems. Many of these initiatives are being linked to a worldwide project called GBIF [73] – Global Biodiversity Information Facility. GBIF intends to set up an interoperable network of biodiversity databases and database management tools that will allow Web users to navigate and query across these databases. Other initiatives are being conducted at smaller scales. Most of these systems are very new, and still under construction. Considerable effort is being applied to creating databases for species' taxonomic descriptions (e.g., [33, 61, 86]), and software on these databases, but still with little help from GIS (e.g., [62]).

Another trend is to process species' spatial distributions using GIS, for a more re-

duced set of species (e.g., [5, 15]). Efficient spatial data management and retrieval, query processing, interface design, and interoperability are among the many problems faced in the design and development of such systems. Spatial databases pose several research and implementation challenges (e.g., [80]). Some of these challenges are motivated by the intrinsic nature of the geographic data – they are location-sensitive and vary in time.

Another difficulty is that the spatial dimension introduces questions related to spatial integrity constraints and spatial query processing [127], involving topological, metric, and directional queries [80]. Our work is not concerned with solving specific problems within the geographic database realm. Rather, we have taken advantage of existing solutions in spatial query processing and combine them with our image processing mechanisms.

A particular issue faced by our approach is that of interoperability. Interoperability problems occur in the GIS context (e.g., [20, 69, 85, 107, 115, 118]). In fact, new geographic applications appear every day, and cover several space-time scales and distinct kinds of objects and phenomena. Moreover, the data are gathered in massive volumes, and proceed from different sources with distinct levels of generalization and incompatible scales.

Several approaches have been discussed to provide geographic systems interoperability and data integration/conversion. Our problem, however, is that of promoting interoperability across systems of different natures – i.e., textual descriptions, image content, and spatial data management. As far as we know, ours is the first proposal that promotes this kind of interoperability.

Image Databases: Image databases (e.g., [71, 117]) combine research on databases and image processing, involving problems that vary from storage issues to friendly interfaces [139]. Images are particularly complex to manage – besides the volume they occupy, retrieval is application- and context-dependent [129]. Even though many other content-based retrieval systems exist [71, 117, 153], they do not take advantage of the component philosophy. Thus, they are not easily amenable to reuse in distinct situations. Our proposal has the advantage of encapsulating CBIR functionality into a DL component, thereby ensuring its reusability and coupling to other DL-based systems.

Digital Libraries There are several DL initiatives that cover topics related to our research. One example is the digital museum of butterflies [83], which aims at building a digital collection of Taiwanese butterflies. This digital library contains 6 modules: XML-based information organization of digitized butterfly collections, content-based image retrieval of butterflies, synchronized multimedia exhibition, compositional FAQ, interactive games regarding butterfly ecosystems, and on-line courseware on butterflies. Queries based on butterfly spatial location are not supported. Another example is floristic digital libraries (FDL) [132]. These are distributed virtual spaces comprising botanical data

repositories and a variety of services offered to library patrons to facilitate the use and extension of existing knowledge about plants. FDL uses an agent-based infrastructure to manage information about taxonomic keys, distribution maps, illustrations, and treatments (morphological descriptions). Content-based retrieval is not supported.

2.5 Conclusions

This chapter presented results of an ongoing project for biodiversity information management, that combines work in image databases with that of geographic distribution of species and their ecosystems. Its originality lies not so much in solving issues in geographic or in image systems, but in providing a solution that combines features from both systems. It relies on a system architecture which extends spatial query processing with retrieval based on image content and textual descriptions, thereby proposing new ways of posing georeferenced queries. In this context, the main challenges to be considered are: the necessity of interaction mechanisms to allow users to easily formulate queries; the difficulty of combining mechanisms of content-based retrieval in image databases and queries of geographical databases; and the complexity of the management of such heterogeneous data. Images, metadata, and maps are stored in databases and are to be retrieved according to a set of predicates based on combining textual and visual descriptors of image content, spatial operators and metadata. A key issue in this architecture is that several query-processing techniques must be investigated, according to user profiles and to the way images and spatial data are preprocessed before being stored.

The solution proposed is based on using new or recently developed DL components. This architecture is easily extensible, and provides users a considerable degree of flexibility in data management. Furthermore, the implementation we provide complies with both digital library (e.g., OAI) and OpenGIS standards (e.g., GML and WFS). Our solution solves many current problems in this kind of system, allowing handling of images, geographic data, and textual information in an integrated fashion. This architecture was conceived to be applied to several domains. In order to show its feasibility, this chapter describes a specific implementation of the architecture to build a fish species biodiversity information system. In particular, this system will be used by students in ichthyology courses at the Department of Fisheries and Wildlife Sciences, Virginia Tech.

Ongoing work concerns the investigation of query optimization techniques to speed up query evaluation across the different sources of evidence. For this part of the work we will take advantage of previous research and development conducted at the University of Campinas in biodiversity query processing [52].

Capítulo 3

A Digital Library Framework for Biodiversity Information Systems

3.1 Introduction

Environmental changes have emerged as an important question in the global agenda. In order to support the design of policies for environmental management and ecosystem balance, it is necessary to get an accurate view of existing conditions, and to understand the complex changes that occur at all levels in the planet. One essential step to creating appropriate scenarios is to collect relevant data about the environment and to develop information systems to manage and derive knowledge from these data. These systems must furthermore combine newly gathered data with historical and legacy information (e.g., from distinct kinds of archives) under homogeneous management. Therefore, scientists concerned with environmental issues must seek support from a large set of systems. This, of course, brings about all kinds of interoperability problems due to system mismatch, data diversity, and variety of user profiles.

One representative example of such problems appears in the context of biodiversity, where expert end-users must contend with at least two kinds of unrelated systems: Biodiversity Information Systems (BIS) and image information systems. The latter involve software that allow users to manage images' content (e.g., patterns, color, texture). In the biodiversity context, they are adopted by scientists for their image archives and to help them identify species.

A BIS (e.g., [5, 15, 16]) is an environmental information system that manages huge sets of geographic data as well as large databases concerning species (e.g., natural history collections, field observation records, experimental data). Geo-related data concern all kinds of geophysical information, provided both by ground surveys and by remote sensing. Most biodiversity information systems are concerned with determining the spatial

distribution of one or more living species, and the spatio-temporal correlations and trends of these distributions. This requires combining data on species (when and where they are observed, by whom and how) with geographic data that characterize the ecosystems where the species are observed. An example of a standard spatial query in a biodiversity system is "Show the areas in Brazil where the plant species *Acacia polyphylla* has been observed". Besides being heterogeneous in nature (encompassing flora and fauna and the geophysical description of their habitats), these data also are heterogeneous in other aspects – such as spatio-temporal granularity or storage format.

Drawings and photos of species also may be used in this context. They are stored apart in the system's data files, and treated as auxiliary documentation, usually retrieved by species' name. Generally, images are accessed only via textual (metadata) queries, without support for content-based image retrieval, e.g., "Show all photos of plant species *Acacia polyphylla*".

If, on the other hand, a scientist starts from incomplete pictorial information – e.g., just a photo of a plant leaf – he/she will have to resort to an image information system to request "Retrieve all database images containing plants whose leaves are shaped like those in the photo". Once likely candidates are identified, the scientist then can continue work by turning to a BIS. Complex biodiversity queries actually may require switching several times across systems.

The goal of the research presented in this chapter is to combine research on image processing, databases, and digital libraries to provide biodiversity researchers with a BIS that provides seamless integration of queries involving both image content and textual data. In such a context, users will just need to provide an image as input (e.g., the photo of a plant leaf) and request the system to "Show the areas in Brazil where the plant species *Acacia polyphylla* coexists with plants whose leaves are shaped like those in the photo".

Our previous work in this direction has concentrated on image metadata, and image processing and analysis techniques for extracting appropriate descriptors from species' images [42, 44, 50], with a prototype implemented. Our present focus is to combine these with digital library (DL) facilities, which offer an organized infrastructure to integrate networked collections of heterogenous data.

To this end, we present a generic digital library architecture for managing heterogeneous data about living beings and their ecosystems. These data involve not only textual and location features, but also images of these beings. A key notion considered is that of a *DL component*, a specially designed software module that encapsulates specific functionality, thereby supporting modularity, flexibility, and reuse in constructing the DL infrastructure. Due to its component-based design, this architecture circumvents the interoperability and system-switching problems discussed. To illustrate the use of this

architecture in a real application, it has been instantiated to support the creation of a BIS for fish species. The goal of that BIS is to help researchers on ichthyology, in particular to identify fish specimen by using search retrieval techniques available in the proposed architecture.

The main contributions of this chapter are the following: (a) a generic architecture for managing heterogeneous collections, based on digital library components, to access heterogeneous biodiversity data sources (text and images), that allows combining text-based and content-based queries in a seamless way; and (b) a new component, for content-based image search, integrated into that architecture.

The rest of this chapter is organized as follows. Section 3.2 characterizes the proposed architecture, including its search components. Section 3.3 describes preliminary experiments conducted to validate the architecture. Section 3.4 briefly comments on related research. Section 3.5 presents conclusions and summarizes ongoing and future work.

3.2 Architecture

This chapter proposes a generic architecture for managing heterogeneous biodiversity data in an integrated fashion. The starting point for the solution is the assumption that the source data are stored in a network of heterogeneous collections organized in a digital library. This architecture takes into account two kinds of collections: image-related and domain-specific databases. In this sense, this architecture can be instantiated for managing data of different domains. For example: for an information system dealing with fish information, the image-related data might include fish photos, while domain-specific collection might contain data about fish taxonomy, morphologic descriptions, etc. For another system handling medicinal plant data, the image-related collection might contain plant photos, while its respective domain-specific collection might include description about plant medicinal properties and known side effects.

3.2.1 Main Modules

Figure 3.1 shows the digital library architecture proposed for managing these heterogeneous collections. This architecture includes a set of search services (service providers) which are executed over heterogeneous data collections (data providers).

This architecture has been implemented by using a set of digital library components which have been developed at Virginia Tech. It uses the Open Archives Initiative (OAI) protocol [94, 116] as a basis for interoperability. OAI is an HTTP- and XML-based protocol for metadata harvesting. It supports digital library interoperability via a two-party model. At one end, data providers use the OAI protocol to publish structured data and

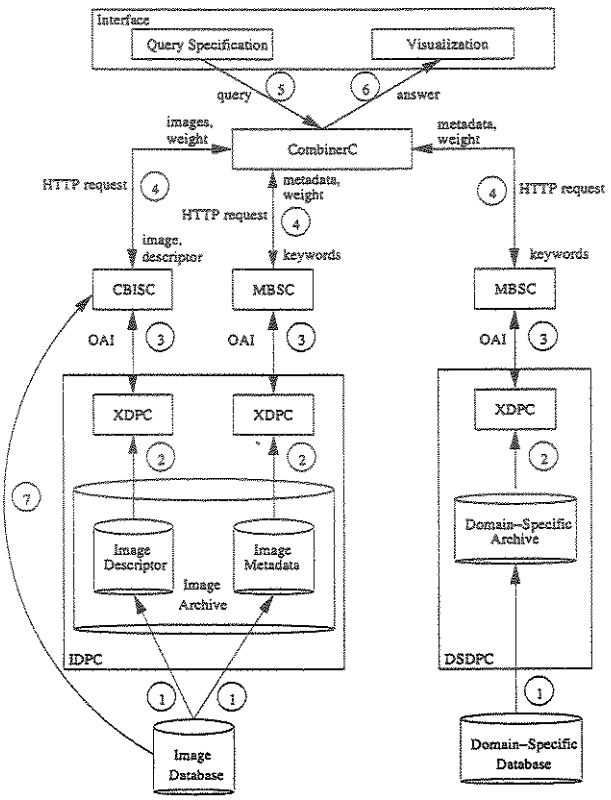


Figura 3.1: System Architecture.

metadata, in various forms. At the other end, service providers use the OAI protocol to harvest the metadata from data providers, to process it, and to add value in the form of services.

The main modules interact as follows. The original data sources are stored in image and domain-specific (e.g., ecosystem) databases. These collections must be pre-processed in order to generate their respective (open) archives (arrows labeled 1 in Figure 3.1): the *Image Archive* for the image collection and the *Domain-Specific Archive* for the domain-specific collection. Archives contain metadata and content descriptors, which speed up retrieval of the original data sources. This process is performed by batch programs that convert the original data sources into XML files. *XML Data Provider Components (XDPC)* are used to publish these XML files as OAI archives (arrows labeled 2), so that these XML files can be accessed through OAI requests (arrows labeled 3). An archive and its respective data provider can be seen as a complex data provider component. The complex data provider for the image collection is named *Image Data Provider Component (IDPC)*. The *Domain-Specific Data Provider Component (DSDPC)* is the complex data provider

for the domain-specific collection.

Search components process queries against these archives. Queries are specified in terms of HTTP requests (arrows 4). A *Metadata-Based Search Component (MBSC)* handles both image metadata and domain-specific information. A *Content-Based Image Search Component (CBISC)* handles image content descriptors expressed in terms of feature vectors, which can be accessed either locally or remotely. In the former case, the *CBISC* can access directly these files (arrow 7). In the latter, they are accessed via the *Image Data Provider Component* (arrow 3).

These search components are activated by the *Combiner Component (CombinerC)*. The Combiner receives a query as input (arrow 5), decomposes it into sub-queries, dispatches them to the search components, combines their results in a suitable way, and then returns a final answer to the interface layer (arrow 6).

The interface layer is not discussed in this chapter. An initial effort to provide users with semantically meaningful result presentations in *CBIR* systems is described in [40]. The following is a description of the other modules.

3.2.2 Data Providers

The *Image Data Provider Component (IDPC)* and the *Domain-Specific Data Provider Component (DSDPC)* are complex components responsible for managing archives by using OAI-compliant XML data providers.

Archives: In this chapter, the term "archive" is used to denote a repository of well-structured stored information; these repositories contain sets of XML files. Two different archives are foreseen in the proposed architecture: *Image Archive* and *Domain-Specific Archive*. The *Image Archive* comprises image metadata and image content descriptors (feature vectors), while the *Domain-Specific Archive* concerns metadata related to a specific domain.

XML Data Provider Component (XDPC): *XMLFile* [145, 159] is an OAI-based component which is used as the XML Data Provider in the proposed architecture. Basically, *XMLFile* is a Perl module that creates an OAI-compliant repository (data provider) to publish a set of XML files as an OAI archive. Its layout and configuration afford a clean separation between the data provider engine, the configuration data, and the data being published. This component does not require any specific metadata format in which the XML files should be encoded.

3.2.3 Search Components

The proposed architecture uses two different search components: a metadata based search component called *ESSEX* and a content-based image search component.

Metadata-Based Search Component (MBSC)

The *ESSEX* search engine [59] is being used as our metadata-based search component. *ESSEX* is a componentized vector-space search engine optimized for digital libraries. *ESSEX* acts as the core portion of an Open Digital Library (ODL [145]) search component, answering requests transmitted through an extended OAI (XOAI) protocol. *ESSEX*, available as open source software, was primarily developed for the CITIDEL (Computing and Information Technology Interactive Digital Educational Library) project [32], and now also is being used in the *PlanetMath* project [123]. In *ESSEX*, all information is indexed in "chunks" associated with field names, where chunks may correspond to XML elements in a metadata record. Its high speed is the result of both keeping index structures in memory and using a background daemon model based on socket communication with the DL application.

Content-Based Image Search Component (CBISC)

The *CBISC* is a new search component we created to support queries based on image content. This component can be used for building special image information systems called *Content-Based Image Retrieval (CBIR) systems*. These systems can be characterized as follows. Assume that we have an image database containing a large number of images. Given a user-defined query pattern (e.g., a query image), retrieve a list of the images from the database which are most "similar" to the query pattern according to the image content (i.e., the objects represented therein and their properties, such as shape, color, and texture). Even though many others content-based retrieval systems exist [11, 71, 117], they do not take advantage of the component philosophy. Thus, they are not easily amenable to reuse in distinct situations. Our proposal has the advantage of encapsulating CBIR functionality into a DL component, thereby ensuring its reusability and coupling to other DL-based systems.

A typical *CBIR* solution requires the construction of **image descriptors**, which are characterized by: (i) an *extraction algorithm* to encode image features into a *feature vector*; and (ii) a *similarity measure* (distance metric) to compute the similarity between features of two images by computing the distance between the corresponding vectors. The similarity measure is a *matching function* (e.g., the Euclidean distance), which gives the degree of similarity for a given pair of images represented by their feature vectors. Usually, the degree of similarity between two images is defined as an inverse function of the distance metric, that is, the larger the distance value, the less similar the images.

Figure 3.2 shows an overview of our *CBISC* component. It receives as input an HTTP request (arrow labeled 1 in Figure 3.2) which specifies a query in terms of the query pattern (query image), chosen descriptor, and kind of query. The *CBISC* starts processing a query

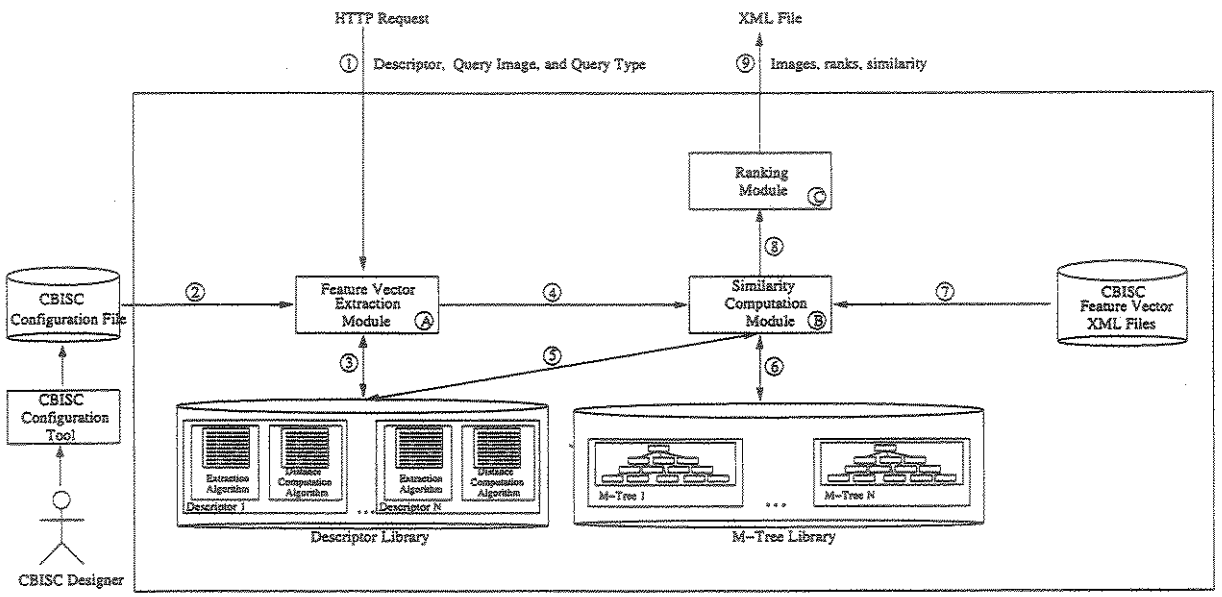


Figura 3.2: *CBISC* architecture.

by extracting a feature vector from the query image (module labeled A in Figure 3.2). This extraction process requires validating the proposed query against the *CBISC* configuration file (arrow 2) and searching for the appropriate *Extraction Algorithm* in the *Descriptor Library* (arrow 3). The validation process is related to check the input query parameters accordingly to the CBISC configuration. For example, it checks if a descriptor defined in the HTTP request is supported by the CBISC or if the input image matches the image type (colorful or binary) used by the image descriptor used in the query.

In the following, the query image feature vector is used to rank the database images according to their similarity (based on a metric distance) to the query image (module B). This step relies on either performing a *Distance Computation Algorithm* (arrow 5) taking into account the feature vectors of all images in the database (arrow 7), or using an appropriate index structure (arrow 6). Images are indexed in the *CBISC* according to their feature vectors by using the M-tree [31] index structure to speed up retrieval and distance computation. The *M-Tree Library* in Figure 3.2 is a repository of M-Trees. Its implementation is based on the eXtensible and fleXible Library (XXL) [21, 54]. Finally, the most similar images are ranked (module C) and the *CBISC* returns an XML file containing this ranked list (arrow 9).

The following text presents the kinds of queries *CBISC* supports and the steps necessary to configure and install the *CBISC*.

Descriptors are typically domain and usage-dependent. Thus, a given image can be associated with very many descriptors. Many *CBIR* methods only support a fixed set of descriptors. *CBISC*, on the other hand, allows progressive extension of the descriptor base.

- **Feature Vectors Extraction**

  Once suitable descriptors have been identified, their extraction algorithms are executed, generating a set of XML files containing the feature vectors for each image. Again, this step is performed prior to component configuration. Figure 3.3 presents an XML schema for the feature vector information. Basically, a feature vector XML file contains information related to the image name, descriptor name, type of feature vector (1D or 2D curve), the feature vectors themselves represented in terms of a curve (double vectors), and their location. A feature vector can be accessed either locally or remotely. In the former case, the *CBISC* can access directly these files (arrow 7 in Figure 3.1). In the latter, they are accessed via the *Image Data Provider Component* (arrow 3 in Figure 3.1).

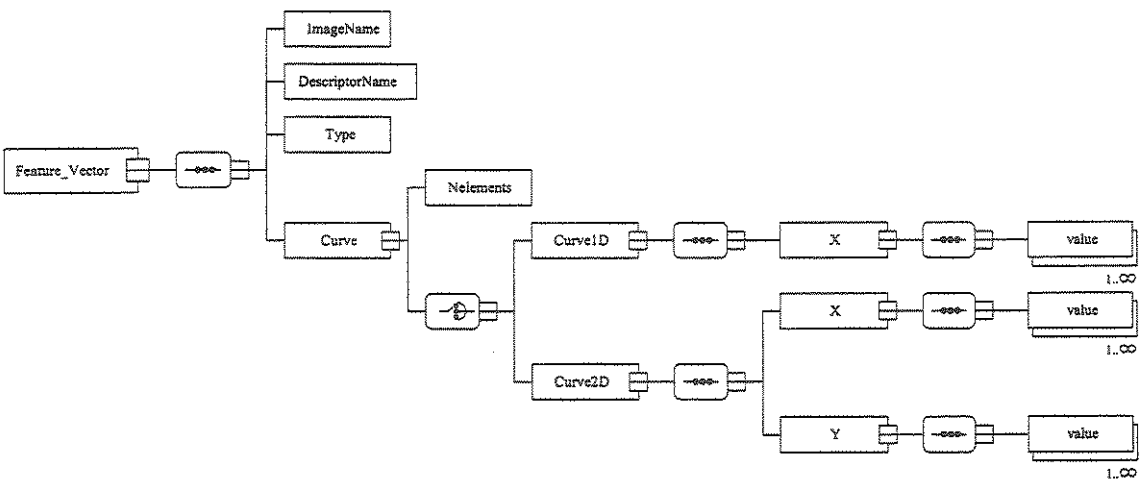


Figura 3.3: Feature vector XML schema.

One of the most important features of the *CBISC* is its flexibility in supporting different kinds of image descriptors. Firstly, the *CBISC* can be configured to perform queries involving different image properties (color, texture, or shape). In this case, it is just required that the extraction algorithm defined in an image descriptor generates a feature vector XML file as specified in Figure 3.3, using the XMLSpy notation [160]. Secondly, the *CBISC* supports extraction algorithms which create either 1D or 2D feature vectors. Thus, 1D feature vectors can be generated for

image descriptors like the *Color Histogram* [148] and the *Contour Multiscale Fractal Dimension* [44] shape descriptor. Similarly, 2D feature vectors can be extracted by, for example, the *Contour Saliences* [50] or the *Curvature Scale Space* [109] shape descriptors.

Figure 3.4 presents an example of a feature vector XML file. In this case, the feature vectors were obtained by applying the image descriptor "Contour Multiscale Fractal Dimension" [44] on image "fish0.pgm". Note that this feature vector is encoded in a 1D curve.

- **CBISC XML Configuration**

  Once the feature vector XML files have been created, the *CBISC* can be configured. Basically, this process involves the creation of an XML configuration file detailing which descriptors are available and the image database related to this component. Figure 3.5 shows the XML schema that defines the *CBISC Configuration XML file*. *DescriptorInformation* includes a list of descriptors that are supported by the *CBISC*. Each descriptor is given in terms of its: name, extraction algorithm, distance computation algorithm, related feature vector size, and location of corresponding feature vector files. Image database information includes the number of images and their location.

```
<?xml version="1.0" encoding="UTF-8"?>
−<feature_vector:Feature_Vector xmlns:feature_vector="http://feathers.dlib.vt.edu/~rtorres/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://feathers.dlib.vt.edu/~rtorres/
    http://feathers.dlib.vt.edu/~rtorres/feature_vector.xsd">
    <feature_vector::ImageName>fish0.pgm</feature_vector:ImageName>
    <feature_vector:DescriptorName> ContourMSFractalDimension <feature_vector:DescriptorName>
    <feature_vector:Type> 1 <feature_vector:Type>
    −<feature_vector:Curve>
        <feature_vector:Nelements> 25 <feature_vector:Nelements>
        −<feature_vector:Curve1D>
            −<feature_vector:X>
                <feature_vector:value> 0.9510525959448239 4192 <feature_vector:value>
                <feature_vector:value> 0.9855121458815461 1995 <feature_vector:value>
                <feature_vector:value> 1.0041549276550782 9986 <feature_vector:value>
                <feature_vector:value> 1.0093103223793751 2441 <feature_vector:value>
                <feature_vector:value> 1.0058378157274110 4426 <feature_vector:value>
                <feature_vector:value> 0.9996517873437800 1835 <feature_vector:value>
                <feature_vector:value> 0.9964170000121828 0747 <feature_vector:value>
                <feature_vector:value> 1.0005341384621639 9108 <feature_vector:value>
                <feature_vector:value> 1.0144805104554643 9042 <feature_vector:value>
                <feature_vector:value> 1.0385244714327943 6048 <feature_vector:value>
                <feature_vector:value> 1.0707932685266490 2248 <feature_vector:value>
                <feature_vector:value> 1.1076428201555308 3838 <feature_vector:value>
                <feature_vector:value> 1.1442544537091177 1370 <feature_vector:value>
                <feature_vector:value> 1.1753678160121783 2360 <feature_vector:value>
                <feature_vector:value> 1.1960510493186684 5774 <feature_vector:value>
                <feature_vector:value> 1.2024088895344982 0420 <feature_vector:value>
                <feature_vector:value> 1.1921365932016856 3482 <feature_vector:value>
                <feature_vector:value> 1.1648425354894063 0552 <feature_vector:value>
                <feature_vector:value> 1.1220849430447841 2218 <feature_vector:value>
                <feature_vector:value> 1.0670985330349558 3177 <feature_vector:value>
                <feature_vector:value> 1.0042248230913544 1270 <feature_vector:value>
                <feature_vector:value> 0.9381055561108777 5851 <feature_vector:value>
                <feature_vector:value> 0.8727520490218962 9230 <feature_vector:value>
                <feature_vector:value> 0.8106643256310066 5476 <feature_vector:value>
                <feature_vector:value> 0.7522426305938187 9515 <feature_vector:value>
            </feature_vector:X>
        </feature_vector:Curve1D>
    </feature_vector:Curve>
</feature_vector:Feature_Vector>
```

Figura 3.4: Example of a feature vector XML file.

A list of predefined descriptors (extraction and distance computation algorithms) is available in a tool we developed to configure the CBISC, called the *CBISC Configuration Tool*, allowing a quick *CBISC* instantiation for a new image collection. Examples include new shape descriptors like the *Contour Multiscale Fractal Dimension* and *Shape Saliences, Beam Angle Statistics - BAS)* [7, 42, 44, 50] and color descriptors, such as the *BIC* [144], and the *Color Histogram* [148]. Common metrics like L1 and L2 (Euclidean distance) also are supported.

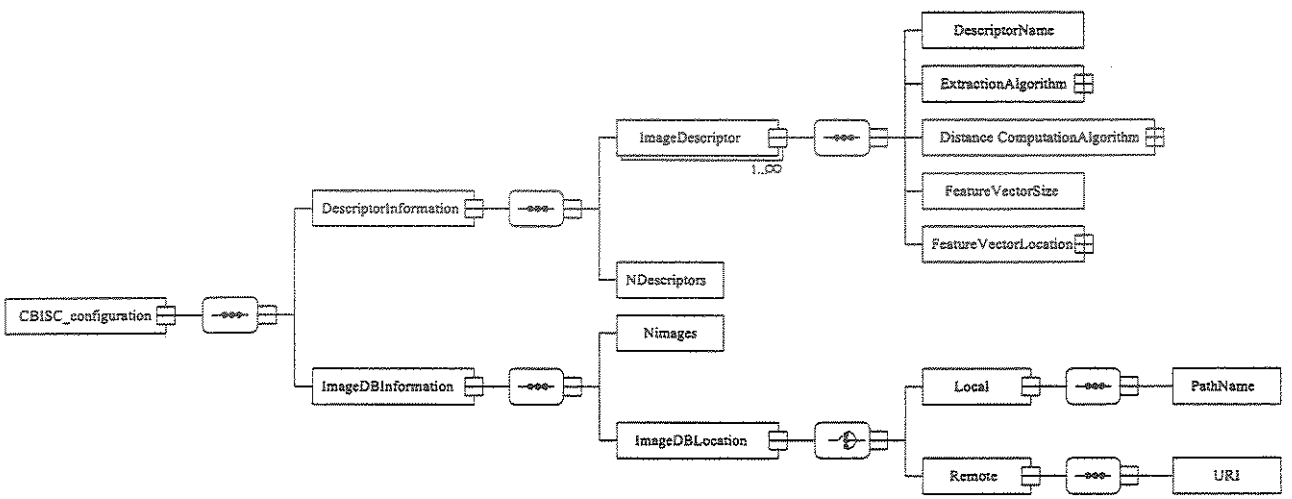


Figura 3.5: XML schema for the CBISC Configuration file.

Figure 3.6 presents a screen shot showing the *CBISC Configuration Tool* developed to support *CBISC* designers in the configuration process.

After the above preliminary steps are performed, the *CBISC Designer* is able to install the *CBISC*. This task also is supported by the *CBISC Configuration Tool*. Basically, this process involves copying feature vectors and algorithms (extraction and distance computation algorithms) either from local directories or from remote sites (by using OAI requests) to *CBISC* main directories. The location of both the feature vector and algorithms are defined in the configuration step.

In fact, *CBISC* flexibility also relies on the support of both locally and remotely defined feature vectors and algorithms. In this sense, a *CBISC Designer* is able to configure a *CBISC*, even without having previous knowledge about the algorithms (descriptors) code. This ease in configuration, and the DL component philosophy, allow BIS designers to easily combine distinct kinds of query features into the system, thereby creating different user-tailored BISs for the same underlying archive base.

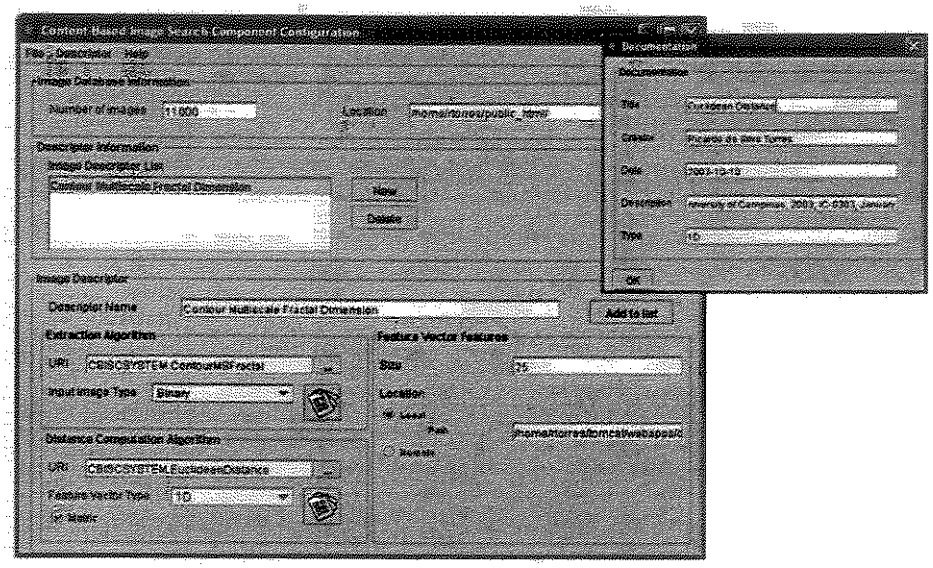


Figura 3.6: *CBISC Configuration Tool* screen shot.

Note that the pre-processing of images into the image descriptors repositories adds scalability and promotes a new, generic way of exposing image archives for creating image-based services.

3.2.4 The Combiner Component

The Combiner component is responsible for combining three different kinds of evidence: content-based retrieved images, image metadata, and domain-specific metadata. Basically, it receives as input a specification in terms of a query pattern (query image) or query terms, decomposes and regroups them into sub-queries, and forwards these resulting sub-queries to the appropriate search component (*CBISC* or *ESSEX*), combines the obtained results (weighted sets) by using an appropriate combination scheme, and returns a ranked list containing the "most" similar objects matching the original specification.

The combiner component has been implemented using search modules found in the *Java MARIAN system* [78]. *MARIAN* is an indexing, search, and retrieval system optimized for digital libraries which has been developed at Virginia Tech. Its search module is based on mapping abstract object descriptions to weighted sets of objects. In this case, the weight of each object in the set serves as a measure of how well that object matches the description.

Given a collection of weighted sets, different searching approaches can be used in the *MARIAN* system to combine them. The most commonly used types of combination include the maximization union and the summative union. The maximization union keeps

only the maximum value of weighted objects that occur in incoming sets. The summative approach, on the other hand, calculates an average of the sums of incoming object sets. Other weighting schemes such as Euclidean distance or sum-of-squares also can be used.

Consider for example, a biodiversity information system which manages fish descriptions (images and textual information). A user might start by providing an image as input (e.g., a photo of an observed fish) and then asking the system to *"Retrieve all database images obtained from 'Randall's tank photos' containing fish with contour shaped like that in the photo, and that are found in the 'Amazon basin' "*. This query deals with three different kinds of evidence: content-based image descriptors (image containing objects shaped like that in the input photo), image metadata (images from "Randall's tank photos"), and domain-specific metadata (species from "Amazon basin").

Given that query, the combiner component proceeds as follows:

1. Parse the original query. This process identifies which search component will be activated and its parameters;
2. Dispatch the query image to the *CBISC* module;
3. Dispatch the expression "Randall's tank photos" to the *ESSEX* search engine which manages image metadata;
4. Dispatch the term "Amazon basin" to the *ESSEX* search engine that manages domain-specific metadata;
5. Each search engine returns XML files containing records which match their respective queries.
6. These XML files are converted into weighted sets, which are combined, by using, for example, the summative union approach;
7. An XML file containing the final answer is returned to the interface layer.

3.3 Experiments

As an illustration of how this generic architecture can be instantiated, we have implemented two Biodiversity Information Systems concerning fish species. The image data consists of fish photos, and the domain-specific data concerns fish and associated habitat descriptions. With these systems, we have carried out experiments to demonstrate the utility of our approach.

3.3.1 Combination of Evidences

The first experiment aimed at evaluating different strategies to combine textual and image content descriptors, to support exploratory searches as the ones described in our motivating examples.

Data Sources

The fish related data were obtained from FishBase [70], an information system available on CD-ROMs, as well as on-line at www.fishbase.org/search.cfm. FishBase covers over 25,000 species of fish from all over the world, including data about taxonomic classification, common names, population dynamics, fish morphology, metabolism, diet composition, trophic levels, food consumption, and predators.

A subset of these data, including 703 species and 932 images, was used in this work. The following text describes the archives managed in this biodiversity information system.

Domain-Specific Archive: The domain-specific archive contains biodiversity metadata on fish and their ecosystems. It includes data about fish taxonomic classification (species, genus, family, and order names), common names, synonyms, ecological features (food items, diet remarks, etc.), morphological descriptions (sexual attributes, type of mouth, type of teeth, etc.) and a list of occurrences around the world.

Image Archive: The image archive contains metadata on fish images, and image descriptors. The main challenge in processing the images has been finding appropriate descriptors for the images, since species' photos are not "well behaved", because they are often taken using live (moving) species instead of more controlled specimens (that are dead and preserved). Therefore, photos that must be used present many irregularities – such as shape distortions – not found in more traditional image databases (e.g., landscapes or artwork). These distortions complicate content-based retrieval. This required a preprocessing step consisting of: image segmentation, reducing image noise, and image binarization.

Accordingly, for experimental purposes the *CBISC* was configured to use the *Beam Angle Statistics (BAS)* [7] shape descriptor.

The image metadata includes the picture name, the related species code (fish ID), the image format, the color type, the picture type, when the picture was obtained, the author name, when the picture data was entered into the FishBase database, general comments, and last modification date (concerning the image).

Experimental Setup

The experiments were intended to evaluate the effectiveness achieved through the combined use of visual and textual features. In this case, we considered each available image as a query image. All images which depict fish belonging to the same species were grouped into the same relevant set. The average number of images in the relevant sets was 1.33. In order to simulate the presence of users, textual search terms were defined randomly for each query. A random attribute was determined, and then a random textual term was extracted from it. This process was performed for both image metadata and domain-specific descriptions.

Two different combination strategies were evaluated: the *maximization union* and the *summative union* (see Section 3.2.4). Only the best results are presented.

Results

Figure 3.7 shows the precision versus recall graphs concerning the use of textual evidence considering: only image metadata (curve named *ESSEX (IM)* in Figure 3.7), only domain-specific information (curve *ESSEX (DS)*), the combination of the textual evidence using the maximization union strategy (curve *ESSEX (IM + DS) MaxUnion*), and finally the combination of textual evidence using the summative union approach (curve *ESSEX (IM + DS) SumUnion*). Note that both combination-based curves present the best results for recall values less than 0.9. From this point on, all curves present a similar behavior. The summative union related curve is better than maximization union one until recall is equal to 0.8. From this point on, this situation is slightly inverted. Note also the low values found for precision. This behavior is due to the low number of elements in the relevant sets.

Figure 3.8 shows the Precision versus Recall graphs for queries involving both the *MBSC* and the *CBISC* search components. Seven different kinds of queries are evaluated: queries considering only the *CBISC* search engine (curve named *CBISC - BAS* in Figure 3.8); the combination of queries on image content and textual information using the maximization union strategy (curves *CBISC + ESSEX (IM) MaxUnion*, *CBISC + ESSEX (DS) MaxUnion* and *CBISC + ESSEX (IM + DS) MaxUnion* for image metadata, domain-specific information, and both together, respectively); and the same combination, now using the summative union strategy (curves *CBISC + ESSEX (IM) SumUnion*, *CBISC + ESSEX (DS) SumUnion*, and *CBISC + ESSEX (IM + DS) SumUnion*. Note that the queries which used the summative union approach yield the best results. In fact, the best result (curve *CBISC + ESSEX (IM + DS) SumUnion*) concerns the combination of the three available sources of evidence, using summative union. The combination strategies involving the maximization union strategy only yield a better behavior (than

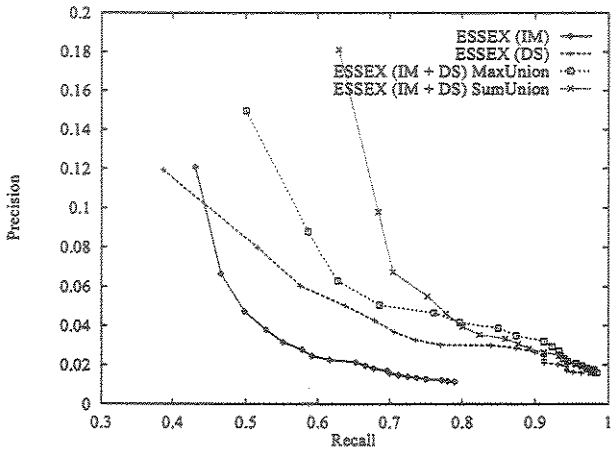


Figura 3.7: Precision versus Recall curves for text based queries.

the curve which considers the use of *CBISC* separately), for recall values between 0.90 and 0.95.

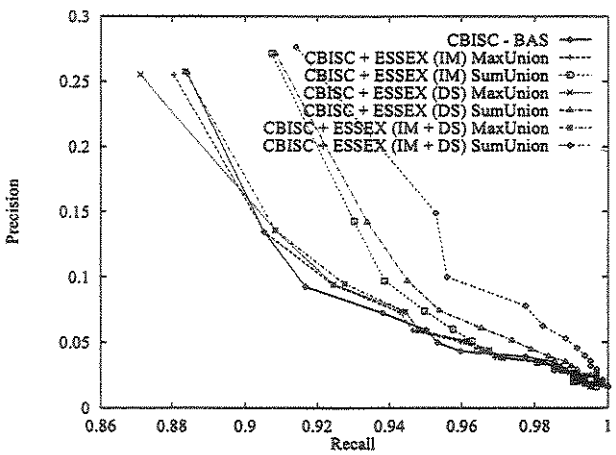


Figura 3.8: Precision versus Recall curves for queries involving both the *MBSC* and the *CBISC* search engines.

The better performance of the summative union method with the three sources further validates our assumption that a combination of several heterogeneous sources of evidence provides enhanced performance, since in this method each source contributes to some degree to the final score, while in the maximization union method only the evidence with the highest score is kept in the final result set.

3.3.2 Fish Identification Process

This experiment involves a user study concerning search retrieval techniques based on the proposed architecture, used to support the fish identification process. The study aims at comparing the effectiveness and the quality of the proposed method versus the traditional key-based approach, using a task-oriented evaluation methodology. This research will potentially have a major impact on the development of new applications for supporting experts during the fish identification process.

The Problem

Given a mixed collection of specimens from a river, ichthyologists face the problem of identifying which fish species are present in that collection. Their aim is to determine which taxonomy classification (e.g, family, genus, species) is appropriate for a given specimen. The traditional approach is based on the use of *keys*. The keys are in the form of dichotomous (two-branched) couplets. Each couplet has two parts (e.g., 1a and 1b); each part of a couplet contains one or more statements. The statements give diagnostic (distinguishing) characteristics (e.g., anatomy, color). All statements in precisely one part of a couplet should fit the fish at hand [89]. Keying involves a sequential comparison of a specimen with a series of paired opposing statements (the parts of the couplets). The process continues, following the applicable statements (those that characterize the fish), until one ends at an identification [89]. For example, Figure 3.9 shows part of the key to families of freshwater fish of Virginia extracted from [89].

Key to Families of Freshwater Fishes of Virginia

1a Paired fins absent; jaws absent, mouth in an oral disk (the disk mostly surrounded by a fleshy hood in larvae); 7 external gill openings present in row behind eye ..... **Lampreys - Petromyzontidae**
1b Paired fins present (at least 1 set); jaws present; 1 external gill opening per side 2
2a Caudal fin heterocercal or abbreviate heterocercal (Figure 5) 3
2b Caudal fin protocercal (Figure 13, Part 2, upper left) or homocercal (Figure 5) 6
3a Snout having a long paddle-like structure; operculum long, flexible, and pointed posteriorly ..... **Paddlefishes - Polyodontidae**
3b Snout lacking a long paddle-like structure; operculum short 4

(...)

Figura 3.9: Part of the key to families of freshwater fish of Virginia [89].

Experimental Setup

Seven student and professor subjects from the Department of Fisheries and Wildlife Sciences at Virginia Tech were recruited. The key selection requirement was expertise in the ichthyology domain. Subjects of any age (over 18) or gender were accepted into the study.

Task: Given a fish specimen, users were asked to identify its corresponding species, genus, and family using the traditional key-based method, and by performing queries on the available system.

Procedure:

- 4 users tried to identify 10 specimens: 5 using the key-based approach and the other 5 using the computer system.
- The other group (3 users) tried to identify the same fish specimens, but used the approaches in the reverse order, concerning the two groups of fish.

Opening Questionnaire: Users were asked to fill out a questionnaire aiming at obtaining information concerning their familiarity with computers and search engines, as well as their expertise in the ichthyology domain and, more specifically, in identifying fish species.

Measured metrics:

- effectiveness: number of correctly identified fish;
- "usability": based on subjective grades (from 1=low to 10=high). With regard to effectiveness and ease of use of the proposed tasks, users were asked to grade both methods. They also were asked to rate their understanding of the computer-assisted fish identification process, both before and after using the tool.
- performance: time spent during the process.

Results

This section presents the experimental results concerning the use of the key-based approach and the computer system for identifying fish species.

On average, users performed 2.5 queries when using the computer system to correctly identify a specimen. 90 queries were performed: 22.2% including only textual terms, 30.0% based on only image content description, and 47.8% using both sources of evidences.

An example of a query including both textual and image descriptor information was: *"retrieve fish descriptions of all fish whose shape is similar to that shown in Figure 3.10, which belong to genus 'notropis', which have 'large eyes' and 'dorsal stripe', and have been observed in both the 'New' and 'Tennessee' rivers".*

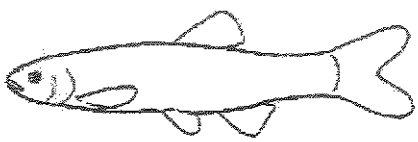


Figura 3.10: Example of shape outline used to define a query.

Figure 3.11 presents a screen shot showing the interface used to define queries by using the fish identification tool. Here, the user can formulate the previous query by selecting on the screen the fish outline that is closest to the request. In addition, text parameters can be entered at the bottom.

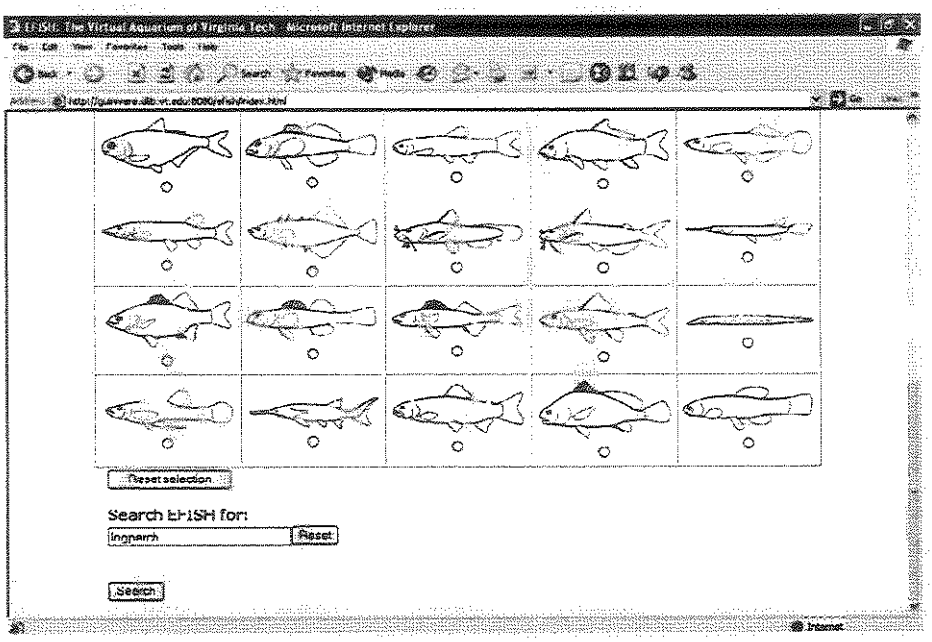


Figura 3.11: Screen shot of the fish identification tool.

Opening Questionnaire: Even though the use of computer is not common for supporting the fish identification process, users are, in general, familiar with computers (five out of seven are "fairly familiar" or "very familiar", while two are "somewhat familiar"). A similar result was found when the users were asked to grade their familiarity with search

Understanding	
before	4.4
after	8.0

Tabela 3.3: Average grade for user understanding of the computer-assisted fish identification process.

Fish Identification Method	
Key	Computer System
6.1	4.1

Tabela 3.4: Average time in minutes required to identify correctly a specimen.

Users also were asked to rate their understanding of the computer-assisted fish identification process, both before and after this experiment. Table 3.3 shows the average grades. This result (improvement from 4.4 to 8.0) confirms that users were able to learn how to use the proposed information system for identifying fish species.

Performance: Table 3.4 shows the average time required to identify correctly a specimen. By using the computer-based approach, user can identify species more quickly than by using the key-based approach (4.1 minutes against 6.1 minutes).

General Comments: In general, users believe that the computer-assisted approach can be very useful to help them identify fish specimen. We list below some of their comments (that relate to our evaluation, and that also can help guide future refinements of our methods):

- "Pictures on computer-based approach were much more helpful than diagrams in key-based approach";
- "The key-based approach is fine to the family level, sometimes, to genus. But it requires dissections, too many subjective judgments to identify species for large families – e.g., Percidae and Cyprinidae. Computer approach (is) thus much more convenient. (It) certainly can get you to the genus and sometimes to species";
- "The best approach is a mix. Have the computer help you through the key by providing lots of pictures, including pictures of fish showing key features in the key...";

- "For some of these species, a key is required. I think that many misidentification could result from the computer-based approach. The computer-based approach could work well with live specimens";
- "If I had more practice with the computer, I may come to prefer that";
- "(The computer system) uses multiple terms for the same thing (...), but the computer does not recognize these as identical. Maybe it would help to create a glossary so user knows which terms to choose";
- "The program is great and can be useful for experts and beginners. I suggest leaving the family names with the form outlines so that people with experience can narrow search results more quickly".

3.4 Related Work

The research described in this chapter differs from related research in the sense that it takes advantage of tailored DL protocols to seamlessly combine textual and content-based retrieval for biodiversity applications. Furthermore, the use of the software engineering notion of *component* ensures appropriate encapsulation of data and procedures, which allow reuse of the components developed in other DL initiatives involving content-based retrieval.

There are some other DL initiatives for the biodiversity domain. One example concerns floristic digital libraries (FDL) [130–132]. These are distributed virtual spaces comprising botanical data repositories and a variety of services offered to library patrons to facilitate the use and extension of existing knowledge about plants. FDLs use an agent-based infrastructure to manage information about taxonomic keys, distribution maps, illustrations, and treatments (morphological descriptions). Content-based retrieval, however, is not supported.

Another example is the Taiwanese digital museum of butterflies, an initiative of National Chi-Nan University and the National Museum of Natural Center [83]. This digital library contains 6 modules: XML-based information organization of digitized butterfly collections, content-based image retrieval of butterflies, a synchronized multimedia exhibition, compositional FAQ, interactive games of an butterfly ecosystem, and on-line courseware on butterflies. Even though XML documents describing butterfly species are indexed and retrieved by a search engine, this digital library does not support queries that combine image content and textual data.

DL efforts that deal with images appear in other domains. An example is the work of Zhu *et al.* [165], which presents a content-based image retrieval digital library that

supports geographical image retrieval. The system manages airplane photos which can be retrieved through texture descriptors. Key goals of the Alexandria Digital Library ADL [141] and its successor (the Alexandria Digital Earth Prototype System (ADEPT) [88] are to build a distributed digital library accessible over the Internet for geographically referenced materials including maps, satellite images, etc., along with their associated metadata. The ADL system applied image-processing techniques to achieve content (texture)-based access to satellite images. Both initiatives, however, have limited support for queries simultaneously involving image content properties and textual data. From the CBIR domain, initiatives like [11, 71, 117] or more recently [24, 122, 153] also supports search of images according to their content information. Even though these systems are showed to be effective, they can not be easily customized for different domains. Firstly, they have a pre-defined and not extensible set of image descriptors. In addition, since they do not take advantage of the component philosophy, they can not be reused and coupled to other information systems. The *CBISC* component presented in this chapter overcomes these limitations.

In the video retrieval domain, Christel *et al.* [29] extract geographic references from videos aiming at improving access to the Informedia Digital Video Library. The available video retrieval process is based on date (when), word occurrences (what), and location information (where), extracted from the narrative and from the text regions in the video segments. Interactive maps are used to display places discussed in a video segment. The user can interact with these maps through toolbar icons that enable zooming in and out, panning, accessing details relevant to the video content, and selecting search areas. Content-based video retrieval is not supported.

Different strategies have been proposed, aiming at supporting the combination of textual information and visual content in the image retrieval proccess [102, 113, 135, 163, 164]. One approach [135, 163] has been to combine textual information with visual contents by using *Latent Semantic Indexing (LSI)* and *Singular Value Decomposition (SVD)* to support image retrieval on the WWW. The combination strategy of Nakagawa *et al.* [113] is based on clustering image objects according to their visual features and mapping the created clusters into related words determined by psychological studies. A different approach is presented in [164]. In this system, the unification of keywords and feature contents is based on a seamless joint querying and relevance feedback scheme. Keyword annotations for each image are converted into a vector which expresses the probability of a determined keyword appearing for a given image. An algorithm for the learning of word similarities during a relevance feedback process also is presented. Finally, Lu *et al.* [102] propose a strategy based on semantic networks and relevance feedback to deduce and utilize the images' content for retrieval.

In contrast to the monolithic-method adopted by the aforementioned solutions, our

approach calculates the combination of textual and visual content in different and autonomous modules. In addition, large systems are either too complex [135, 163] to be easily configured for a new domain, or rely on search process techniques (relevance feedback, word similarity learning, content semantic definitions) [102, 113, 164], which are not available in the proposed architecture. Note that new combination strategies easily can be adopted in the proposed architecture. New combiners just have to follow the HTTP-based communication protocol presented here.

The XML schema adopted in our Content-Based Image Search Component to encode feature vectors (see Figure 3.3) is similar to MPEG-7 [26, 138] solutions recently proposed to describe multimedia data content. MPEG-7, for example, includes a *Description Definition Language (DDL)*, which defines representation data structures such as matrices and arrays, to encode feature vectors of different visual features. Furthermore, the MPEG-7 initiative also standardizes a set of descriptors applied to images and/or videos [18, 105]. Current work investigates both the use of MPEG-7-based tags to define feature vectors and the incorporation of MPEG-7 image descriptors into the CBISC descriptor set.

3.5 Conclusions

Interoperability has been a central research area in the digital library domain [119]. The OAI protocol has been used to promote interoperability solutions for different digital libraries initiatives [77, 94]. Following this trend, this chapter presented an OAI-based generic digital library architecture for integrated management of image descriptors and textual information. The solution proposed is based on using DL components which are mostly new or recently developed. This architecture is easily extensible, and provides users a considerable degree of flexibility in data management. To illustrate our claim that this architecture can be applied to several domains, this chapter describes its application in building two biodiversity information systems on fish species. This solution solves many current problems in this kind of system, allowing handling of images and textual information in an integrated fashion.

A new Content-Based Image Search Component was presented that supports queries on image collections. Since this component is based on OAI principles, it provides an easy-to-install search engine to query images by content. It can be readily tailored for a particular collection by a designer (a domain expert), who carries out a clearly defined set of pilot experiments. It supports the use of different types of image descriptors (metric and non-metric; color, texture and shape descriptors; with different data structures to represent feature vectors), which can be chosen based on the pilot experiment, and then easily combined to yield improved effectiveness. Besides, it encapsulates a metric index structure to speed up the search process, that can be easily configured for different image

collections.

We also have validated the proposed architecture for two applications, dealing with different collections on fish species. Firstly, we performed experiments concerning the combination of textual and image content information. Preliminary results show that when both textual and visual information are used in the image retrieval process, results are, in general, better than those achievable using only visual or textual information. On the average, better results were found by using the summative union combination strategy. Secondly, we have evaluated the use of the proposed architecture to help fish experts in the process of fish identification. Results show that the fish identification process based on the information system built on top of the proposed architecture is more effective, easier and less time consuming than that based on the traditional key-based approach. Consequently, we plan to provide students in ichthyology courses at Department of Fisheries and Wildlife Sciences at Virginia Tech with an information system based on the proposed architecture aiming at supporting the process of learning new fish species.

Ongoing work concerns the instantiation of the proposed architecture in other domains. For instance, we are trying to combine queries on image content with textual description in the archeology domain [60]. In this case, the image collection comprises photos of archaeological artifacts (e.g., pottery, coins, etc) and the domain-specific collection corresponds to both archaeological site information and artifact descriptions. Preliminary experiments confirm the reusability of the components developed. Future work includes performing user experiments to evaluate the different combination strategies which can be used by the *Combiner Component*. We also intend to evaluate other image descriptors [44, 109, 144, 148] in the combination process.

Capítulo 4

A Graph-based Approach for Multiscale Shape Analysis

4.1 Introduction

In pattern recognition and related areas, shape is an important characteristic to identify and distinguish objects [100]. The shape variations expressed with respect to a given scale, named *multiscale shape representation*, provide even more information about the objects. In this context, shape descriptors have been used to encode such representations into signatures (i.e., feature vectors). In practice, objects belong to certain semantic categories, each category defines a class, and the problem consists of grouping the objects that belong to a same class. The main challenge here is to find out "good signatures" to perform such a task successfully.

This chapter presents the advantages of computing two recently proposed shape descriptors, multiscale fractal dimension and contour saliences [36, 42], using the *image foresting transform* (IFT)— a graph-based approach to the design of image processing operators [63–65, 67, 101]. In this case, the shape descriptors are obtained from the multiscale shape representations created by the IFT. The multiscale fractal dimension [36, 39] is a new concept, which copes with many serious drawbacks in current methods [110, 120] for numerical estimation of fractal dimension. The multiscale fractal dimension of a shape is computed based on the Euclidean distance transform (EDT) of its pixels. The EDT of these pixels is also related to their geometric Voronoi diagram [126], where each pixel defines an influence zone (*discrete Voronoi region*) composed by its closest image pixels. The saliences of a shape are computed based on the areas of the discrete Voronoi regions of its higher curvature pixels within a narrow band around the shape [36]. This approach allows the quantification of the curvature values at points (center of pixels) where the analytical curvature would be infinite. The IFT provides the simultaneous computation

of the EDT and the discrete Voronoi regions in time usually proportional to the number of pixels [65], being more efficient than the method proposed in [36]. The present chapter also introduces improvements in the multiscale fractal dimension and contour saliences computations. The original approach for multiscale fractal dimension suffers from undesirable oscillations on the fractal curve, and the location of higher curvature points along the contour for saliences computation is very sensitive in the case of intricate and complex shapes. The oscillation problem is solved using polynomial regression. The relation between the salience points of the contour and of its internal and external skeletons— an important concept introduced in [97]— is used to locate the higher curvature points along the contour, considerably improving the robustness of the contour saliences computation. This relation is obtained in a direct way using the IFT framework. The contour saliences descriptor is also redefined to include point location and salience value along the contour and a special distance metric, which make it possible to reach high effectiveness in shape recognition.

The proposed descriptors are compared with single fractal dimension, two classical (Fourier descriptors [154] and moment invariants [84]), and two recently published shape descriptors (Curvature Scale Space (CSS) [1, 109] and Beam Angle Statistics (BAS) [6, 7]) in regarding to the following aspects: *compact-ability* and *separability*. The compact-ability of a descriptor indicates its invariance to the object variations within a same class, while the separability indicates its discriminatory ability between objects that belong to distinct classes. In other words, a descriptor is considered "good" when it creates compact clusters far away from each other, for all classes in the corresponding feature space. This condition should be sufficient for the success of any suitable classification method.

This chapter starts by presenting an overview of the IFT in Section 4.2. The IFT is used to obtain two types of shape representation: *multiscale contours by exact dilations* and *multiscale skeletons by label propagation*, as described in Section 4.3. We use the former to estimate multiscale fractal dimension and the later to locate the salience points along the contour in Section 4.4. Section 4.5 gives a formal definition of compact-ability and separability, evaluates the proposed shape descriptors, and discusses the main results of this work. We present the conclusion and our current research on shape descriptors in Section 4.6.

4.2 Image Foresting Transform

The *image foresting transform* (IFT) is a recent approach to the design of image processing operators based on connectivity [63–65, 67, 101]. The IFT reduces image partition problems based on a given seed set to the computation of a *minimum-cost path forest* in a directed graph, whose nodes are the pixels and whose arcs are defined by an *adjacency*

relation between pixels. A path in this graph is a sequence of adjacent pixels. The cost of a path is determined by a suitable *path-cost function*, which usually depends on local image properties along the path— such as color, gradient, and pixel position. For suitable path-cost functions, the IFT assigns to each image pixel a minimum-cost path from the seed set, such that the union of those optimum paths form an oriented forest spanning the whole image. The nodes of each rooted tree in the forest are composed by pixels that are "more closely connected" to its root pixel than to any other seed in some appropriate sense. The IFT assigns to each pixel three attributes: its predecessor in the optimum path (predecessor map P), the cost of that path (cost map C), and the corresponding root (root map R) or some label associated with it (label map L).

For a given set S of seed pixels, the IFT can provide the simultaneous computation of the Euclidean distance transform in the cost map C and of the discrete Voronoi regions in the root map R [67]. This operator asks for an Euclidean adjacency relation A and a path-cost function f_{euc} defined for any path $\pi =< p_1, p_2, ..., p_n >$ in the graph as:

$$q \in A(p) \Longrightarrow (x_q - x_p)^2 + (y_q - y_p)^2 < \rho^2, \tag{4.1}$$

$$f_{euc}(\pi) = \begin{cases} (x_{p_n} - x_{p_1})^2 + (y_{p_n} - y_{p_1})^2, & \text{if } p_1 \in S, \\ +\infty, & \text{otherwise.} \end{cases} \tag{4.2}$$

where ρ is the adjacency radius and (x_{p_i}, y_{p_i}) are the (x, y) coordinates of a pixel p_i in the image. Note that, the main idea is to find for every image pixel p_n a path $P^*(p_n)$ from a seed pixel $p_1 \in S$, such that $f_{euc}(P^*(p_n))$ is minimum. The exact Euclidean distance transform will depend on the appropriate choice of ρ, as demonstrated in [67]. However, for most practical situations involving 8-connected curves, such as contours and skeletons, $\rho = \sqrt{2}$ is enough [65]. Algorithm 1 below presents an IFT procedure with f_{euc}.

Algorithm 1:
Input: An image I, a set S of seed pixels in I, and an Euclidean adjacency relation A;
Output: An optimum-path forest P, and the corresponding cost map C and root map R.
Auxiliary Data structures: A priority queue Q.

1. For all pixels p of the image I, set $C(p) \leftarrow +\infty$;
2. For all $p \in S$, set $P(p) \leftarrow nil$, $R(p) \leftarrow p$, $C(p) \leftarrow 0$, and insert p in Q;
3. While Q is not empty, do
   3.1. Remove from Q a pixel $p = (x_p, y_p)$ such that $C(p)$ is minimum;
   3.2. For each pixel $q = (x_q, y_q)$ such that $q \in A(p)$ and $C(q) > C(p)$, do

3.2.1. Set $C' \leftarrow (x_q - x_{R(p)})^2 + (y_q - y_{R(p)})^2$, where $R(p) = (x_{R(p)}, y_{R(p)})$ is the root pixel of p;
3.2.2. If $C' < C(q)$, then
3.2.2.1. If $C(q) \neq +\infty$, then remove q from Q.
3.2.2.2. Set $P(q) \leftarrow p$, $C(q) \leftarrow C'$, $R(q) \leftarrow R(p)$, and insert q in Q.

Note that, the IFT algorithm is essentially Dijkstra's shortest-path algorithm [13, 55, 72, 111], slightly modified to multiple sources and general path-cost functions. Its correctness for weaker conditions that are applied to only optimum paths in the graph is presented in [67].

4.3 Multiscale Shape Representation

A shape can be represented along a range of scales spanning from coarse to fine. If the shape is to be used as an invariant indicator of an object in a scene in which the viewing distance is variable, a multiscale structure is necessary to relate various views, thereby making the representation invariant with respect to the viewing distance [25]. The IFT with f_{euc} allows efficient computation of *multiscale contours by exact dilations* and *multiscale skeletons by label propagation* [65].

4.3.1 Multiscale Contours by Exact Dilations

Given a set S of points, represented in terms of their Cartesian coordinates (x, y), its exact Euclidean dilation by a radius r, henceforth represented as S_r, is defined as being the union of all disks of radius r centered at each of the points in S. Observe that, this definition is valid for both discrete and continuous objects. Subsequent dilations of a given shape by increasing values of r create a family of progressively simplified instances of the original shape, as illustrated in Figure 4.1 for a contour.

Multiscale contours by exact dilation result from Algorithm 1, where the pixels of the original shape (contour) are taken as the seed set S. Each instance of the multiscale shape is obtained by thresholding the cost map C at a given squared Euclidean distance value. The higher the threshold value, the more simplified the shapes become, with smaller details being progressively removed as the threshold increases.

4.3.2 Multiscale Skeletons by Label Propagation

Given a contour with N pixels, its internal skeleton is defined as the geometric location of the centers of maximal disks contained in the contour [17, 92]. A similar definition is

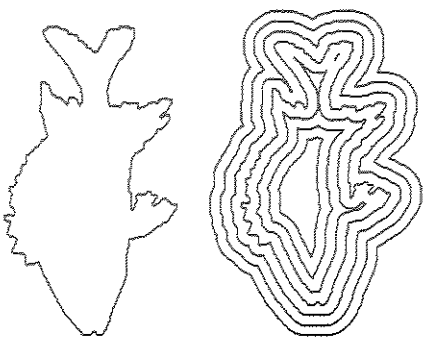


Figura 4.1: (a) A contour of a fish and (b) multiscale contours by exact dilation.

valid for the external skeleton.

Algorithm 1 applied to the contour creates a root map R. Multiscale skeletons can be computed from R if each contour pixel p (root) is assigned to a subsequent label value $\lambda(p)$, varying from 1 to N, while circumscribing the contour (Figure 4.2a). A label map L can be created by computing $L(R(p))$ to each image pixel p (Figure 4.2b). A more efficient way, however, is to propagate the labels of the contour pixels during Algorithm 1. In this case, the labeling function λ is used in step (2), when the contour pixels are inserted in Q, and the label map L is created similarly and simultaneously to the root map R. A difference image D results from the label map L by computing the following for each pixel p inside and outside the contour (Figure 4.2c):

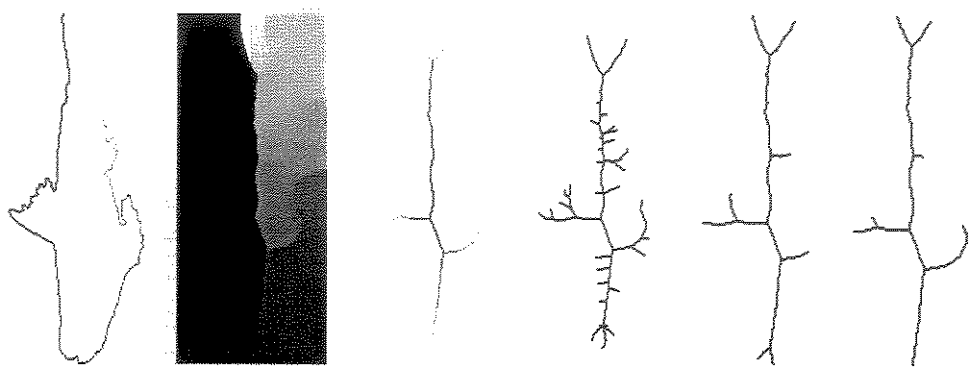
$$D(p) = \max_{\forall q \in A_4(p)} \{\min\{\delta(p,q), N - \delta(p,q)\}\}, \tag{4.3}$$

where $\delta(p,q) = L(q) - L(p)$ and $A_4(p)$ is the set of pixels q that are 4-neighbors of p. The difference image represents the multiscale internal and external skeletons by label propagation [38, 63, 65]. One-pixel wide and connected skeletons can be obtained by thresholding the difference image at subsequent integer values (Figure 4.2d-f). The higher the threshold value, the more simplified the skeletons become, with smaller details being progressively removed as the threshold increases.

It is important to observe that Equation 4.3 corrects the original Equation, reported in [63, 65], as pointed out in [66].

4.4 Shape Descriptors

This section presents the process of creating shape descriptors from the multiscale shape representations presented in Section 4.3.



(a) (b) (c) (d) (e) (f)

Figura 4.2: Multiscale skeletonization by label propagation. (a) Labeled contour, (b) label map, (c) difference image, and (d-f) skeletons at three different scales.

4.4.1 Multiscale Fractal Dimension

While the topological dimension is restricted to integer values, fractal dimension allows fractionary values. Disseminated by Mandelbrot [104], fractal dimension provides an interesting means for characterizing the self-similarity (or self-affinity) of abstract and real objects, being closely related to the concept of power-laws. A particularly intuitive and useful definition of fractal dimension is the Minkowski-Bouligand dimension [152], which is here introduced in terms of the following example. Let the shape under analysis be represented in terms of the set S of the Cartesian coordinates of each of its elements, and let S_r be its dilation by r (see Section 4.3.1). Let $A(r)$ be the area of the respective dilated version of the shape, i.e. S_r. The Minkowski-Bouligand fractal dimension, hence F, is defined as

$$F = 2 - \lim_{r \to 0} \frac{log(A(r))}{log(r)}. \tag{4.4}$$

In other words, the fractal dimension descriptor in this case (i.e. considering a two-dimensional space) is a number within $[0, 2]$. It should be borne in mind that F assumes perfect self-similarity of the shape for small spatial scales, i.e. for r close to 0, which is never verified for real data. Indeed, while shapes in nature can exhibit an infinite degree of detail as one moves into the microscopic scales, the self-similarity along these scales is not preserved for an infinite interval. For instance, a fern leaf presents just a few orders (3 or 4) of self-similarity. The situation is even more complicated for experimental data, where the finite resolution of the acquisition device contributes further to limit the small scale detail.

In spite of such limited fractality observed for real objects, the standard numerical procedure for estimating fractal dimensions involves linear interpolating the logarithm curve of the area ($A(r)$) in terms of dilating radius, computing the angular coefficient ($A'(r)$) of this line and taking F as $F(r) = 2 - A'(r)$ (see Figures 4.3a and 4.3b). Observe that the area values $A(r)$ for each logarithm of the dilation radius r can be simply obtained by computing the accumulated histogram of the cost map of the IFT with f_{euc}. Therefore, it is obtained from the multiscale contours by exact dilations (Section 4.3.1).

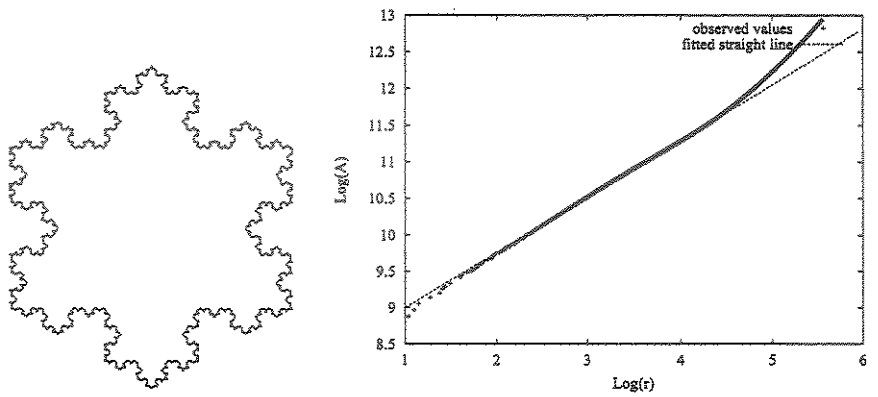


Figura 4.3: (a) An object similar to the Koch star, whose fractal dimension is known as about 1.26 ($= \frac{log4}{log3}$). (b) The logarithmic area function. By taking 2 minus the inclination of the fitted straight line, the fractal dimension obtained is about 1.23.

Although the deviations of shapes from perfect self-similarity seriously undermine the aforementioned experimental method, several practical applications of the fractal dimension have been reported in the literature (e.g., [149]). Fractal dimensions have been considered as features useful for expressing the area coverage and the "complexity" of shapes ranging from neurons [39] to heartbeat dynamics [4]. In the particular case of the Minkowski-Bouligand dimension, the value of F provides an interesting indication of how much the shape constrains its own dilation. Therefore, simple shapes, such as the point or the straight line impose relatively less constraints to their own dilation and consequently have smaller fractal dimension values than those of an intricate curve in the plane.

In order to address the subjectivity implied by the choice of the interval over which the logarithmic curve is interpolated and to fully take into account the limited self-similarity exhibited by the geometry of real shapes, the concept of *multiscale fractal dimension* was recently reported [36]. This approach involves taking into the infinitesimal limit the previous concept of linear interpolation [110, 120], which naturally leads to the estimation of the *derivative* of the logarithmic area function. Therefore, the multiscale fractal dimen-

sion becomes a function of the spatial scale rather than a single scalar global value. By expressing the fractality explicitly in terms of the spatial scale, this new measure provides a richer description of the self-similarity of the analyzed shapes along the spatial scales. The derivative function therefore becomes completely independent of the choice of the spatial scale interval adopted for interpolation.

The approach presented here fits a polynomial curve by regression to the logarithmic area function from which the sought derivatives can be immediately obtained. One important advantage of this approach is to be free of the undesirable oscillations often found in the derivative estimation of sampled curves. Note that, the commonly used fractal dimension can be understood as a particular case of the multiscale dimension when the adjusting polynomial is linear. The multiscale fractal dimension is obtained whenever the degree of the polynomial is greater than one. In the examples of this chapter, the multiscale fractal dimension is represented by a polynomial of degree nine. In this work the multiscale fractal dimension descriptor is represented by a vector of 50 sample points of this polynomial. The polynomial degree and the vector size were determined through a set of experiments. These experiments showed that vectors containing more than 50 sample points do not improve the results. Two multiscale fractal vectors are compared using the L_2 metric.

Figure 4.4 illustrates the concept of multiscale fractal dimension with respect to the contour in Figure 4.3a. Observe that the maximum value of the curve in Figure 4.4b is close to 1.26, which is the actual fractal dimension of the Koch triadic curve (up to three digits).

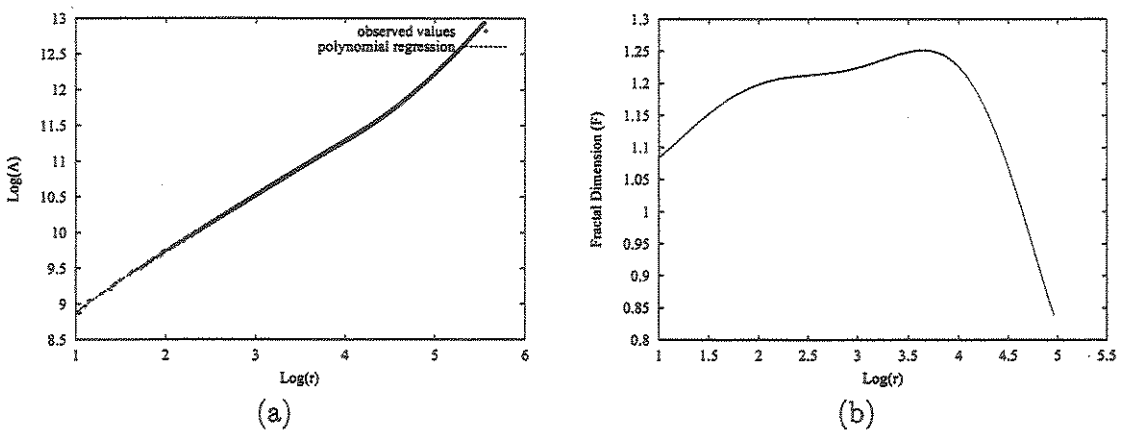


Figura 4.4: (a) The $log \times log$ curve of the areas of each exact dilation radius for Figure 4.3a. (b) The multiscale fractal dimension of its contour.

4.4.2 Shape Saliences

The storage of the area evolution for each point of the shape also provides perspectives for shape descriptions. The influence areas of higher curvature points, namely *salience points* [36], are expected to be greater than the influence areas of the other points of the shape. Moreover, in the case of a contour, the influence area of a convex point (point A) is greater outside the contour than inside, and the other way around for a concave point (point B, see Figure 4.5). The influence area A of each salience point relates to the aperture angle θ, illustrated in Figure 4.5, by the formula:

$$Area = \frac{\theta \times r^2}{2}, \tag{4.5}$$

where r is a dilation radius. Costa et al. [36] proposed to estimate the salience points by thresholding the influence areas, computed for low values of r (e.g., $r = 10$). The influence area A of each pixel belonging to a shape (contour or skeleton) can be simply obtained from the histogram of the root map R created by Algorithm 1, restricted to pixels p where $C(p) \leq r^2$. This approach, however, misses important salience points in opposite parts of the shape which come close to each other. It has otherwise been particularly effective for skeletons and for simple contours, such as convex polygons, but it fails in finding the salience points of more complex and intricate contours. A robust approach to solve this problem for contours is described next.

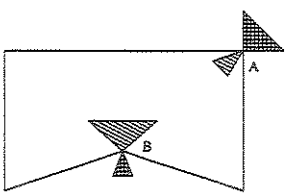


Figura 4.5: Internal and external influence areas of a convex (A) and a concave (B) point.

For a given contour, multiscale internal and external skeletons are first obtained by label propagation as described in Section 4.3.2. For small scales (e.g., 5% of the maximum label difference $N - 1$), each salience point of the internal skeleton corresponds to one convex point of the contour and each salience point of the external skeleton corresponds to one concave point of the contour [97] (see Figure 4.6). Let L, R_i, R_e be the label and root maps resulting from Algorithm 1 applied to the contour with label propagation, to the internal skeleton, and to the external skeleton, respectively. The influence areas of each point of the skeletons are determined based on the histogram of R_i and R_e, restricted to pixels within a narrow band around the skeletons (e.g., $r = 10$). The salience points of the skeletons are those with influence area greater than an area threshold obtained by

setting $\theta = 70$ in Equation 4.5. In order to locate the salience points along the contour from the salience points of the skeletons, the algorithm uses the label map L as follows. Note that, Equation 4.3 essentially assigns to each pixel inside and outside the contour the maximum length of the shortest contour segment between two roots equidistant to the pixel. Figure 4.7 illustrates this situation for a salience point c in the skeleton, which is related to a salience point a in the contour. The difference value $D(c)$ is the length of the segment $\overline{dab}$. Suppose the root pixel of c is b, the point a can be reached from the point c by skipping $\overline{dab}/2$ pixels in the anti-clockwise orientation along the contour starting from b. Similarly, the point a can be found from c through d following the clockwise orientation, when d is the root pixel of c. The method needs only to determine which is the root pixel, either b or d. If the contour pixels are labeled in clockwise orientation, the root pixel of c will be b whenever $\delta(p,q) > (N - \delta(p,q))$ in Equation 4.3 for $L(q) = L(d)$ and $L(p) = L(b)$. Otherwise, the root pixel of c will be d for $L(q) = L(b)$ and $L(p) = L(d)$. The same rule is applied for the external skeleton.

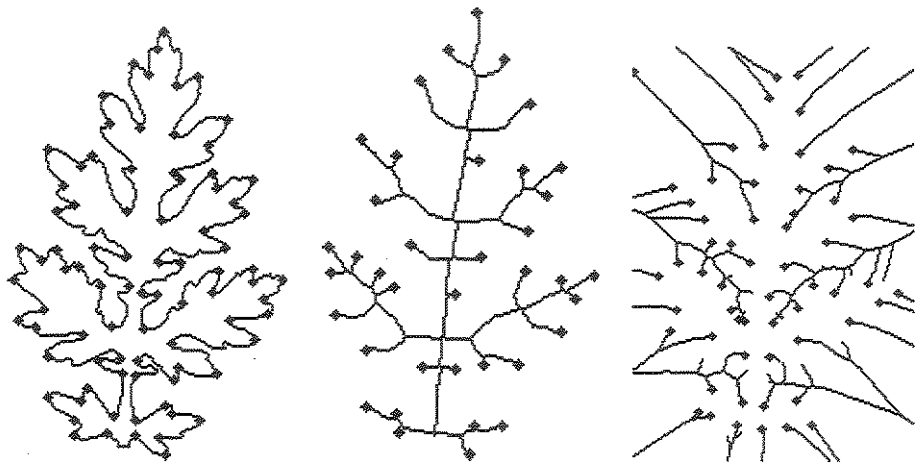


Figura 4.6: (a) Saliences of the contour of a leaf and (b-c) saliences of its internal and external skeletons.

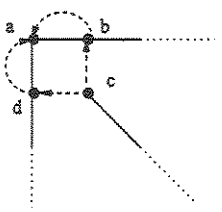


Figura 4.7: Relation between skeleton and contour saliences.

The location and the influence area of the salience points along the contour represent important local and global information for shape analysis. The influence areas (salience values) are obtained from the histogram of L restricted to a narrow band around the contour (e.g., $r = 10$). They are signed negative for concave points and positive for convex points. An arbitrary point of the contour is taken as reference point and the algorithm computes the relative position of each salience point with respect to the reference point along the contour. Finally, a *contour saliences descriptor* is defined as two vectors of the same size: one with the salience values and the other with the relative position of the salience points along the contour. Note that the dimension of these vectors may be different for different contours as well as the reference points. A special algorithm has been designed for matching this descriptor between two contours taking into account these differences. This algorithm is described in Section 4.5.3.

Figure 4.8 illustrates the contour saliences descriptor for a polygon. The contour of the polygon, its reference point (A), the internal and external skeletons, and the respective salience points are indicated in Figure 4.8a. Figure 4.8b indicates the salience values of the vertices of the polygon by their relative position along the contour.

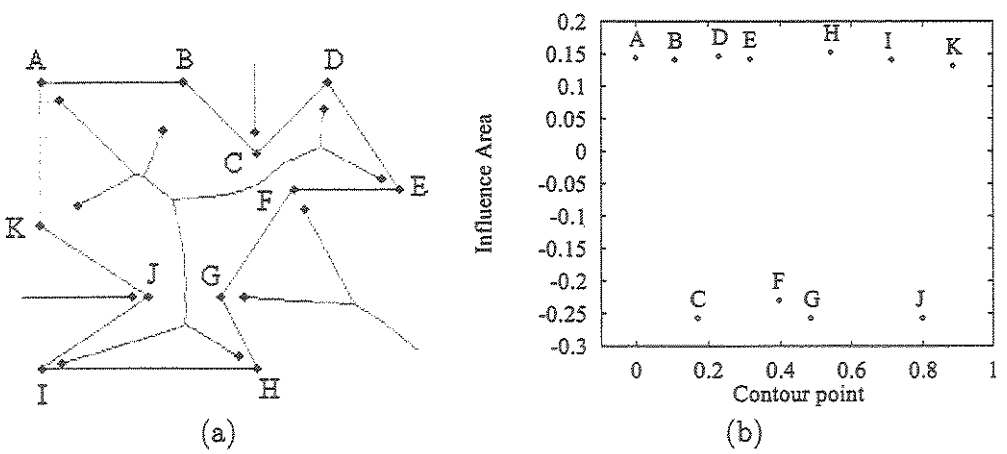


Figura 4.8: (a) Contour and skeletons of a polygon, where salience points are indicated by dots. (b) The salience values of the vertices of the polygon by their relative position along the contour.

4.5 Evaluation

For classification purposes, a descriptor is considered more effective than another one when it increases the number of correctly classified objects. These objects are organized into classes according to some semantic criterion. A "good" shape descriptor should

represent different classes of objects by compact clusters of points separated from each other in the corresponding feature space. These aspects ask for two concepts: *compact-ability* and *separability*. The compact-ability of a descriptor indicates its invariance to the object characteristics that belong to a same class, while the separability indicates its discriminatory ability between objects that belong to distinct classes. They evaluate the "goodness" of a description independent of the classification method. Moreover, the separability determines the effectiveness of the descriptor independent of the compact-ability. However, the compact-ability gives an idea of how the separability may be affected if the number of classes increases.

The shape descriptors presented in this chapter are evaluated with respect to compact-ability and separability in the context of a specific application. This application aims at designing and implementing an architecture for integrating image and spatial data for biodiversity information management. This architecture has been specified in a generic way, but its implementation is being carried throughout for the specific case of fish species.

One thousand and one hundred fish contours were obtained from the database available at [136] for the experiments. Figure 4.9 shows some examples of fish contours and their respective skeletons together with the respective salience points.

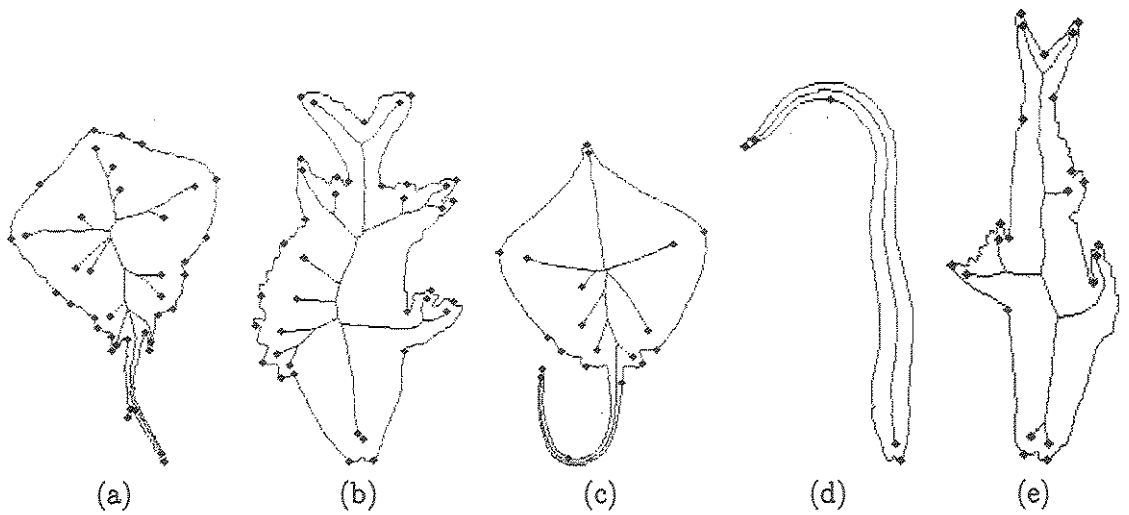


Figura 4.9: Fish images used for descriptor evaluation. The concave points were determined through the salience points of the external skeleton, not shown in the figure.

Since there is no semantic definition of classes for the fish contours in this database, each class is defined as consisting of 10 different manifestations of each contour by rotation and scaling. Then, the problem consists of 1,100 classes with 10 shapes each, totalizing 11,000 contours. In this case, compact-ability becomes the invariance to possible rotation and scaling of a given shape, and separability becomes the discriminatory ability of a

descriptor among the 1,100 classes of the database.

A precise definition of compact-ability and separability, the matching algorithm for contour saliences, and the experiments are presented in the next sections.

4.5.1 Compact-ability

Let Σ be a set (database) of κ shapes organized in classes. The compact-ability $\phi_D(C)$ of a descriptor D for a given class C in Σ is defined as:

$$\phi_D(C) = 1 - \frac{\sum_{\forall i,j \in C} \Delta_D(i,j)}{|C|^2 \max_{\forall i,j \in C}\{\Delta_D(i,j)\}}, \tag{4.6}$$

where $|C|$ is the number of shapes in the class C, $\Delta_D(i,j) = Distance(D_i, D_j)$ and D_i is the value of D for shape i. Note that this measure is normalized with respect to the maximum distance between a pair of shapes, considering all shapes in the class C.

The compact-ability of a given descriptor is the average of the normalized compact-abilities of this descriptor over all classes in Σ, i.e.:

$$\Phi_D = \frac{\sum_{\forall C \in \Sigma} \phi_D(C)}{|\Sigma|}, \tag{4.7}$$

where $|\Sigma|$ is the number of classes in the set Σ.

4.5.2 Separability

Let Σ be a set (database) of κ shapes organized in classes. The separability $\psi_D(C)$ of a descriptor D for a given class C is defined as follows. An arbitrary shape r_C is taken as reference for the class C and the distances $\overline{\Delta_D(r_C, i)} = \frac{\Delta_D(r_C,i)}{M}$, where $M = \max_{\forall i \in \Sigma, \forall r_C}\{\Delta_D(r_C, i)\}$, is computed for all shapes i in Σ.

The distance range is quantized in a certain number of values from x to 1.0 with intervals of dx (e.g., $x = 0.02$ and $dx = 0.02$). Let $\eta_{r_C}(x)$ be the number of shapes, whose distance from the reference shape is less than or equal to x ($\overline{\Delta(r_C, i)} \leq x$) and do not belong to the class C ($i \notin C$). For each distance value from x to 1.0, the separability $\psi_D(C)$ of a descriptor D with respect to class C is defined as:

$$\psi_D(C) = 1 - \frac{\eta_{r_C}(x)}{\kappa}, \tag{4.8}$$

These separability values define a multiscale curve of separability for the class C along x. The separability of a descriptor D is defined as the average of the multiscale separability over all classes, i.e.:

$$\Psi_D = \frac{\sum_{\forall C \in \Sigma} \psi_D(C)}{|\Sigma|} \tag{4.9}$$

4.5.3 Matching Algorithm for Contour Saliences

Whenever two contours of a same object appear in different positions, they should be represented by the same salience points along the contour. Therefore the pairwise comparison between objects using contour saliences requires matching between contours.

The contour saliences descriptor considered in the current work preserves the salience values of the points along the contour and their relative position regarding to a reference point. These characteristics encode a lot of information about the shape. The reference point is used only for correction of the relative positions after the matching. The matching algorithm proposed in this chapter is based on the matching algorithm proposed to match Curvature Scale Space (CSS) images presented in [1, 109].

Let $S_A = \{(u_{A1}, s_{A1}), \ldots, (u_{An}, s_{An})\}$ and $S_B = \{(u_{B1}, s_{B1}), \ldots, (u_{Bm}, s_{Bm})\}$ be two salience descriptors of contours A and B, where (u_{Ai}, s_{Ai}) stands for the i^{th} salience value s_{Ai} at the position $u_{Ai} \in [0, 1]$ along A.

1. Create $S'_A = \{(u'_{A1}, s'_{A1}), \ldots, (u'_{An}, s'_{An})\}$ and $S'_B = \{(u'_{B1}, s'_{B1}), \ldots, (u'_{Bm}, s'_{Bm})\}$ by sorting S_A and S_B according to the decreasing order of salience values.

2. Create a list L containing a pair of matching candidates points from S'_A and S'_B. A pair $((u'_{Ai}, s'_{Ai}), (u'_{Bj}, s'_{Bj}))$ belongs to the list L if $|s'_{Ai} - s'_{Bj}| \leq 0.2 s'_{A1}$. A pair $((u'_{Bj}, s'_{Bj}), (u'_{Ai}, s'_{Ai}))$ belongs to the list L if $|s'_{Bj} - s'_{Ai}| \leq 0.2 s'_{B1}$.

3. For each pair of matching candidates in the form $P_{ij} = ((u'_{Ai}, s'_{Ai}), (u'_{Bj}, s'_{Bj}))$ in L, find the shift parameter α as $\alpha = u'_{Ai} - u'_{Bj}$. Shift S_A salience points by α, yielding $S''_A = \{(u''_{A1}, s''_{A1}), (u''_{A2}, s''_{A2}), \ldots, (u''_{An}, s''_{An})\}$.

4. The distance d between S''_A and S_B is given as:

$$d = \sum_{k=1}^{\min\{n,m\}} d_k,$$

where

$$d_k = \begin{cases} \sqrt{(u''_{Ak} - u_{Bk})^2 + (s''_{Ak} - s_{Bk})^2}, & \text{if } |u''_{Ak} - u_{Bk}| \leq 0.2 \\ s''_{Ak} + s_{Bk}, & \text{otherwise.} \end{cases}$$

Finally, if $n \neq m$, it is added to d the height s of the not matched points.

5. Repeat the steps 3 and 4 by considering matching candidate pair in the form $P_{ij} = ((u'_{Bj}, s'_{Bj}), (u'_{Ai}, s'_{Ai}))$ in L.

6. Select the lowest distance d as the distance between S_A and S_B.

4.5.4 Experiments

Clearly, the multiscale fractal dimension is not scale invariant. In order to reduce this problem the contours have been first normalized according to their diameter. Even though the area thresholding method can be applied to locate the salience points of an external skeleton. These points may not correspond to relevant concave salience points along the contour. The reason is that the external skeleton may present spurious branches due to rotation and scaling of the contour, and the salience points of those branches should not be considered. In fact, they can be eliminated for distinct contours by varying the area threshold in Equation 4.5 (see Figure 4.9). However, a fixed area threshold (i.e. $\theta = 70$ and $r = 10$ in Equation 4.5) may affect the performance of the contour saliences descriptor with the concave points are considered. Therefore, the experiments used only the convex salience points along the contour.

Table 4.1 shows the set of implemented shape descriptors. The proposed descriptors (*D2* and *D3*) are compared with the single fractal dimension (*D1*), two classical descriptors (Fourier descriptors (*D4*) and moment invariants (*D5*)) and two recently published shape descriptors (Curvature Scale Space (D6) and Beam Angle Statistics (D7)). Many versions of these methods have been presented, but this work considers their conventional implementations.

Descriptor Id	Descriptor Name
D1	Fractal Dimension
D2	Multiscale Fractal Dimension
D3	Contour Saliences
D4	Fourier Descriptors
D5	Moment Invariants
D6	Curvature Scale Space (CSS)
D7	Beam Angle Statistics (BAS)

Tabela 4.1: List of evaluated descriptors.

Fourier Descriptors: The Fourier descriptors of a contour consist of a feature vector with the 126 most significant coefficients of its Fourier Transform using the method described in [79, 106]. The Euclidean distance was used to measure the similarity between two Fourier-descriptors vectors.

Moment Invariants: For Moment Invariants, each object was represented by a 14-dimensional feature vector, including two sets of normalized Moment Invariants [57, 84], one from the object contour and another from its solid silhouette. Again, the Euclidean

distance was used to measure the similarity between different shapes represented by their Moment Invariants.

Curvature Scale Space Descriptor (CSS): The CSS descriptor is a shape descriptor, adopted in MPEG-7 standard [18], which represents a multiscale organization of the curvature zero-crossing points of a planar curve. The extraction algorithm of the CSS descriptor is described in [1, 109]. A special matching algorithm is necessary to compare two CSS descriptors (e.g., the algorithm presented in Section 4.5.3). The experiments used a C version of the Matlab prototype presented in [108].

Beam Angle Statistics (BAS): The BAS [6, 7] is a novel shape descriptor which has been compared with several others [14, 30, 91, 96, 99, 109]. In [7], it is shown that BAS functions with 40 and 60 samples outperform all of them. The experiments of the present chapter used the BAS descriptor with 60 samples. Basically, the BAS descriptor is based on the *beams* originated from a contour pixel. A beam is defined as the set of lines connecting a contour pixel to the rest of the pixels along the contour. At each contour pixel, the angle between a pair of lines is calculated, and then the shape descriptor is defined by using the third-order statistics of all the beam angles in a set of neighborhood systems. The BAS algorithm is presented in [6, 7]. The similarity between two BAS moment functions is measured by an optimal correspondent subsequence (OCS) algorithm as shown in [7].

4.5.5 Experimental Results

Initially, the multiscale fractal dimension of a contour (*D2*) was compared with its single fractal dimension (*D1*). Figure 4.10 shows that the multiscale version of the fractal dimension descriptor presents the best separability curve.

Figure 4.11 shows the separability curves of the proposed descriptors (*D2* and *D3*), against the Fourier descriptors (*D4*), the moment invariants (*D5*), the CSS (*D6*) and the BAS (*D7*). Observe that the Contour Saliences (*D3*), CSS (*D6*) and BAS (*D7*) present equivalent performance for search radii less than 18% of their maximum distance. From this point on, the BAS's separability curve (*D7*) decreases quickly, being worse than the separability of the Multiscale Fractal Dimension (*D2*) and of the Fourier descriptors (*D4*) for search radii above 25%. This behavior indicates that the BAS descriptor is neither robust nor effective for search radii greater than 25%. The Multiscale Fractal Dimension (*D2*) has a better separability curve than Fourier descriptors (*D4*), moment invariants (*D5*) and BAS (*D7*) for search radii between 25% and 40%. Its performance, however, decreases, being lower than Fourier descriptors (*D4*) for search radii greater than 40%. The most relevant result is that the Contour Saliences descriptor (*D3*) has the best separability curve considering all search radii. Although its performance is equivalent to

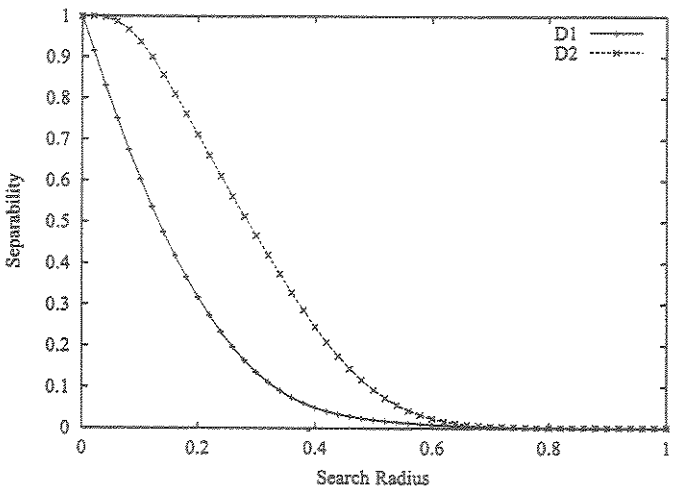


Figura 4.10: Multiscale separability diagrams for the shape descriptors based on fractal dimension.

Descriptor Id	Compact-ability
D1	0.93
D2	0.97
D3	0.70
D4	0.77
D5	0.97
D6	0.73
D7	0.95

Tabela 4.2: Compact-ability values of the evaluated descriptors.

the famous CSS descriptor (*D6*) for search radii less than 30%, it is more robust and more effective for higher search radii.

Table 4.2 presents the compact-ability values of the evaluated shape descriptors. The higher values were found for single fractal dimension (*D1*), multiscale fractal dimension (*D2*), moment invariants (*D5*), and BAS (*D7*), while the contour saliences (*D3*) presented the lowest value. Fortunately, the compact-ability 0.70 of *D3* can not be considered sufficiently low to interfere in its separability, even considering a database with 1,100 classes. According to these experiments, *D3* is more effective than the others.

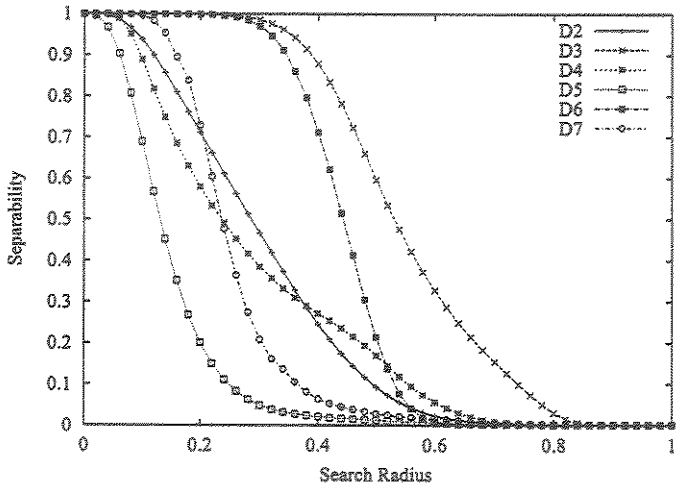


Figura 4.11: Comparison of the proposed descriptors with the Fourier descriptors, invariant moments, CSS, and BAS.

4.6 Conclusion

This chapter has presented two effective shape descriptors, multiscale fractal dimension and contour saliences, using the framework of the IFT. The presented method to compute multiscale fractal dimension is more efficient [65] and robust than the one published in [36], since the undesirable oscillations commonly found in Fourier-based approaches have been eliminated here by the use of polynomial regression. The location of salience points along a contour was computed in a direct way using the IFT framework to exploit the relation between the contour and its skeletons [97]. This method is more robust and efficient than the approach presented in [36]. Moreover, the chapter redefines the contour salience descriptor to include point location and salience value along the contour and a special distance metric.

The multiscale fractal dimension and the contour saliences were also extensively evaluated for the first time, using a database with 1,100 classes and 11,000 contours. Their "goodness" (compact-ability and separability) have been showed by using as references the single fractal dimension, two classical (Fourier descriptors [79, 154] and moment invariants [57, 84]) and two recently proposed shape descriptors (Curvature Scale Space [1, 109] and Beam Angle Statistics [6, 7]). The underlying ideas of compact-ability and separability may not be totally new concepts, however this chapter has presented an original way to compute them, especially the multiscale separability.

The experiments showed that the contour saliences descriptor was the most effective (with the best separability curve). This is certainly a breakthrough result considering

that the experiments have taken into account recent descriptors and a database with 1,100 classes. The Multiscale Fractal Dimension is competitive with BAS and Fourier Descriptors in terms of separability and compact-ability, but it is less effective than the CSS and the Contour Saliences. This may indicate that the normalization procedure was not effective to make the Multiscale Fractal Dimension totally scale independent. In [7], the BAS descriptor was shown to be more effective than CSS for the MPEG-7 Core Experiments shape-1. The experiments with separability showed the opposite result. Note that one cannot say that a descriptor is better than another without taking into account several application domains.

Current work concerns to solve the scale-dependency problem of the multiscale fractal dimension and to incorporate concave points in the composition of the contour saliences descriptor. In view of that, we are investigating a special distance metric for the multiscale fractal dimension and an automatic area thresholding method to avoid salience points of the spurious branches of the external skeleton. We are also interested in validating the proposed descriptors for other application domains. In special, we are currently considering applications in content-based image retrieval.

Capítulo 5

Contour Salience Descriptors for Effective Image Retrieval and Analysis

5.1 Introduction

Recent technological improvements in image acquisition and storage have supported the dissemination of large databases, where the design of information retrieval systems based on image properties becomes a challenge [139]. In these Content-Based Image Retrieval (CBIR) systems, image properties are usually represented by shape, color, and texture of objects/regions within the image. A CBIR system essentially consists of an image database, a *descriptor*, and a data structure for image indexation. The descriptor is a pair, *feature vector* and *distance metric*, used for image indexation by similarity. The feature vector subsumes the image properties and the distance function measures the dissimilarity between two images with respect to their properties. Each image can be interpreted as a "point" in the underlying metric space, where similar images form groups of points. For given user-defined specification or pattern (e.g., shape sketch, query image), the CBIR system aims at retrieving groups of similar images which are relevant to the query (*effectiveness*) as fast as possible (*efficiency*). Clearly, the efficiency of the system depends on the indexing structure (e.g., a Metric Access Method [31, 151]) and on the complexity of the distance function, while its *effectiveness* is solely related to the ability of the descriptor in representing distinct groups of relevant images as far as possible in the metric space. That is, different descriptors define different CBIR systems with distinct degrees of effectiveness, where the goal of research is to find the descriptor with maximum effectiveness for a given application. The descriptors are also important in image analysis, where the groups of relevant images form classes or patterns for recognition [56]. The

present chapter is mainly concerned with shape descriptors and their effectiveness for image retrieval and analysis.

Costa *et al.* [36] proposed the use of *shape saliences* for object representation. The saliences of a shape are defined as the maximum influence areas of its higher curvature points, considering a *narrow band* in both sides of the curve and the Voronoi regions of its points. A contour point, for example, is considered convex when its influence area is greater outside than inside the contour, and concave otherwise. The narrow band is used to reduce as much as possible cross-influence of opposite parts of the curve, which come close to each other. Torres *et al.* [42] presented a more efficient way to compute shape saliences using the *image foresting transform* [67] and a contour salience descriptor for image retrieval [50] and analysis [44]. In both works, the contour salience descriptor was compared with several other shape descriptors, including the popular *curvature scale space* [1, 109] and the recently proposed *beam angle statistics* [6, 7]. However, the contour salience descriptor never considered concave salience points, because its effectiveness was very sensitive to the precise location of these points. This work solves the problem, incorporating concave points to the contour salience descriptor. In addition, it proposes another shape descriptor based on the *salience values* of contour segments.

The methods use the image foresting transform to compute the salience values of contour pixels and to locate salience points along the contour by exploiting the relation between a contour and its internal and external skeletons [97]. The contour salience descriptor consists of the salience values of salient pixels and their location along the contour, and on a heuristic matching algorithm as distance function. The contour is also divided into a fixed number of segments and the influence areas of their pixels inside and outside the contour are used to compute *segment saliences*. The segment salience descriptor consists of the salience values of contour segments and an optimal matching algorithm as distance function.

The chapter describes the computation of shape saliences using the image foresting transform in Section 5.2. Section 5.3 provides a detailed description of the algorithm to locate salient contour pixels via multiscale skeletonization. The new contour and segment salience descriptors are presented in Section 5.4 and compared with the convex contour saliences, curvature scale space, and beam angle statistics in Section 5.5. Section 5.6 states the conclusion and discusses the current research on CBIR systems.

5.2 Shape Saliences

The algorithm proposed by Costa *et al.* [36] to determine shape saliences is based on the concept of *Exact Dilation with Label Propagation (EDLP)*. The EDLP of a given labeled seed set S assigns to each image pixel t a value $C(t)$ and a label $L(t)$, which are the

minimum Euclidean distance between t and S (Euclidean distance transform) and the label of its closest pixel in S (discrete Voronoi regions), respectively.

The EDLP algorithm can take contour pixels as seeds and determine the influence areas of each seed as the areas of its discrete Voronoi regions inside and outside the contour. The influence areas of higher curvature points, namely *salience points*, are expected to be greater than the influence areas of other contour pixels. Moreover, the influence area of a convex point (points A, B, D, and E in Figure 5.1) is greater outside than inside the contour, and the other way around is true for a concave point (point C in Figure 5.1). The influence area of each salience point relates to the aperture angle θ, illustrated in Figure 5.1, by the formula:

$$Area = \frac{\theta \times r^2}{2}, \tag{5.1}$$

where r is a dilation radius. Costa *et al.* [36] proposed to use as *salience value* of a contour point the maximum influence area between the areas computed outside and inside the contour for a low value of r (e.g., 10), in order to avoid cross-influence of opposite parts of the contour which come close to each other. They also suggested to locate the salience points along the contour by thresholding their salience values (i.e. $Area \geq \frac{\theta \times r^2}{2}$, for some value of θ).

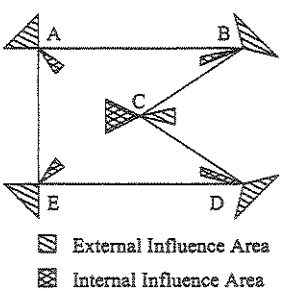


Figura 5.1: Internal and external influence areas of convex (A, B, D, and E) and concave (C) points.

5.2.1 Shape Saliences by Image Foresting Transform

Costa *et al.*'s algorithm [36] can be more efficiently implemented (in time proportional to the number of pixels) by using the Image Foresting Transform (IFT) [42]— a graph-based approach to the design of image processing operators based on connectivity [64, 65, 67, 101].

The IFT reduces image partition problems based on a given seed set to the computation of a *minimum-cost path forest* in a directed graph, whose nodes are the pixels

and whose arcs are defined by an *adjacency relation* between pixels. The cost of a path in this graph is determined by an application-dependent *path-cost function*, which usually depends on local image properties along the path — such as color, gradient, and pixel position. For suitable path-cost functions, the IFT assigns to each image pixel a minimum-cost path from the seed set, such that the union of those optimum paths form an oriented forest spanning the whole image. The nodes of each rooted tree in the forest are composed by pixels that are "more closely connected" to its root pixel than to any other seed, in some appropriate sense. The IFT assigns to each pixel three attributes: its predecessor in the optimum path (predecessor map P), the cost of that path (cost map C), and the corresponding root (root map R) or some label associated with it (label map L).

For a given set S of seed pixels, the IFT can provide the simultaneous computation of the Euclidean distance transform in the cost map C and of the discrete Voronoi regions in the root map R [67]. This operator asks for an Euclidean adjacency relation A and a path-cost function f_{euc} defined for any path $\pi =< p_1, p_2, ..., p_n >$ in the graph as:

$$q \in A(p) \Longrightarrow (x_q - x_p)^2 + (y_q - y_p)^2 \leq \rho^2, \tag{5.2}$$

$$f_{euc}(\pi) = \begin{cases} (x_{p_n} - x_{p_1})^2 + (y_{p_n} - y_{p_1})^2, & \text{if } p_1 \in S, \\ +\infty, & \text{otherwise,} \end{cases} \tag{5.3}$$

where ρ is the adjacency radius and (x_{p_i}, y_{p_i}) are the (x, y) coordinates of a pixel p_i in the image. Note that, the main idea is to find for every image pixel p_n a path $P^*(p_n)$ from a seed pixel $p_1 \in S$, such that $f_{euc}(P^*(p_n))$ is minimum. The exact Euclidean distance transform will depend on the appropriate choice of ρ, as demonstrated in [67]. However, for most practical situations involving 8-connected curves, such as contours and skeletons, $\rho = \sqrt{2}$ is enough [65]. Algorithm 1 below presents the IFT procedures for f_{euc}.

Algorithm 1:
Input: An image I, a set S of seed pixels in I, and an Euclidean adjacency relation A;
Output: An optimum-path forest P, and the corresponding cost map C and root map R.
Auxiliary Data structures: A priority queue Q.

1. For every pixel p of the image I, set $C(p) \leftarrow +\infty$;
2. For every $p \in S$, set $P(p) \leftarrow nil$, $R(p) \leftarrow p$, $C(p) \leftarrow 0$, and insert p in Q;
3. While Q is not empty, do
   3.1. Remove from Q a pixel $p = (x_p, y_p)$ such that $C(p)$ is minimum;

3.2. For each pixel $q = (x_q, y_q)$ such that $q \in A(p)$ and $C(q) > C(p)$, do

3.2.1. Set $C' \leftarrow (x_q - x_{R(p)})^2 + (y_q - y_{R(p)})^2$, where $R(p) = (x_{R(p)}, y_{R(p)})$ is the root pixel of p;

3.2.2. If $C' < C(q)$, then

3.2.2.1. If $C(q) \neq +\infty$, then remove q from Q.

3.2.2.2. Set $P(q) \leftarrow p$, $C(q) \leftarrow C'$, $R(q) \leftarrow R(p)$, and insert q in Q.

Note that, the IFT algorithm is essentially Dijkstra's shortest-path algorithm [55], slightly modified to multiple sources and general path-cost functions. Its correctness for weaker conditions that are applied to only optimum paths in the graph is presented in [67].

A natural extension of this algorithm to compute contour saliences consists of obtaining one histogram of the resulting root map for each side of the contour, restricted to a small neighborhood of the curve in order to eliminate the cross-influence of its opposite parts. Each bin of the histograms indicates the area of influence of the respective root inside (or outside) the contour. The root is classified as *convex*, when the external area is greater than the internal area, and otherwise as *concave*.

As in the original approach [36], a point of the curve is classified as salient by thresholding its maximum influence area [42]. This approach, however, may miss important salience points when opposite parts of the contour come too close to each other, even for a small radius r in Equation 5.1. It has otherwise been particularly effective for skeletons and for simple contours, such as polygons, but it fails in finding the salience points of more complex and intricate curves. Torres *et al.* [44, 50] have proposed a partial solution for this problem, which is described next.

5.3 The Use of Skeletons for Contour Saliences

First, multiscale skeletons [65] are computed for the contour (Section 5.3.1), and one internal skeleton and one external skeleton are chosen by thresholding the multiscale skeletons. Second, the internal and external skeleton saliences are found similarly to as described in the previous section (Section 5.3.2). The location of the contour saliences are determined by relating the salience points of the internal skeleton to convex contour points and the salience points of the external skeleton to concave contour points (Section 5.3.3).

5.3.1 Multiscale Skeletonization

Given a contour with N pixels, its internal skeleton is defined as the geometric location of the centers of maximal disks contained in the contour [92]. A similar definition is valid

for the external skeleton.

Algorithm 1 applied to the contour creates a root map R. Multiscale skeletons [65] can be computed from R if each contour pixel p (root) is assigned to a subsequent label value $\lambda(p)$, varying from 1 to N, while circumscribing the contour (Figure 5.2a). A label map L can be created by computing $L(R(p))$ to each image pixel p (Figure 5.2b). A more efficient way, however, is to propagate the labels of the contour pixels during Algorithm 1. In this case, the labeling function λ is used in step (2), when the contour pixels are inserted in Q, and the label map L is created similarly and simultaneously to the root map R. A difference image D results from the label map L by computing the following for each pixel p inside and outside the contour (Figure 5.2c):

$$D(p) = \max_{\forall q \in A_4(p)} \{\min\{\delta(p,q), N - \delta(p,q)\}\}, \tag{5.4}$$

where $\delta(p,q) = L(q) - L(p)$ and $A_4(p)$ is the set of pixels q that are 4-neighbors of p. The difference image represents the multiscale internal and external skeletons by label propagation [38, 65]. One-pixel wide and connected skeletons can be obtained by thresholding the difference image at subsequent integer values (Figures 5.2d-f). The higher the threshold value, the more simplified the skeletons become, with smaller details being progressively removed as the threshold increases.

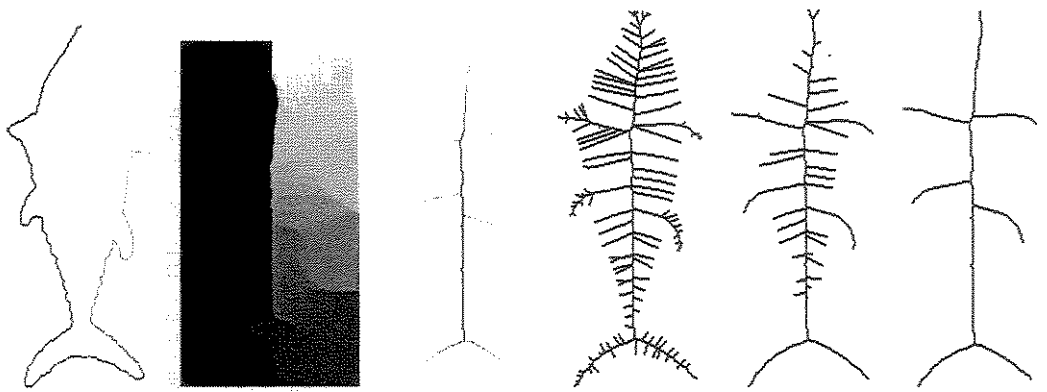


Figura 5.2: Multiscale skeletonization by label propagation inside a contour. (a) Labeled contour, (b) label map, (c) difference image, and (d-f) internal skeletons at three different scales.

5.3.2 Skeleton Saliences

For small scales (low thresholds – e.g., 5% of the number N of contour pixels), each salience point of the internal skeleton corresponds to one convex point of the contour and each

salience point of the external skeleton corresponds to one concave point of the contour (see Figure 5.3). The salience points of the skeletons are determined similarly to as described in Section 5.2.1 by taking the skeleton points as seed pixels and executing Algorithm 1 for each skeleton separately. For a small dilation radius ($r = 10$), the histogram of the root map gives the influence areas of each skeleton point. The salience points of the skeletons are those with influence area greater than the area threshold obtained by setting $\theta = 70$ in Equation 5.1.

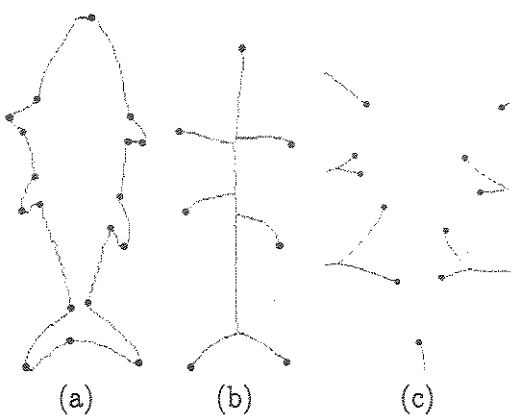


Figura 5.3: (a) Salience points of the contour of a fish and (b-c) salience points of its internal and external skeletons.

5.3.3 Contour Saliences Via Skeletons

The relation between the contour and its internal and external skeletons [97] is directly obtained by applying Algorithm 1 to the contour [44, 50]. Equation 5.4 assigns to each pixel inside and outside the contour the maximum length of the shortest contour segment between two roots equidistant to that pixel according to the cost map. Figure 5.4a illustrates this situation for a salience point c of the skeleton, which is related to a salience point a of the contour. The difference value $D(c)$ is the length of the segment $\overline{dab}$. Suppose b is the root pixel of c, point a can be reached from point c by skipping $\overline{dab}/2$ pixels in the anti-clockwise orientation along the contour, starting from b. Similarly, point a could be found from c through d following the clockwise orientation, when d is the root pixel of c. The method only needs to determine which is the root pixel, either b or d. If the contour pixels are labeled in clockwise orientation, the root pixel of c will be b whenever $\delta(p,q) > N - \delta(p,q)$ in Equation 5.4 for $L(q) = L(d)$ and $L(p) = L(b)$. Otherwise, the root pixel of c will be d for $L(q) = L(b)$ and $L(p) = L(d)$. The same rule is applied for the external skeleton. Figures 5.4b-c illustrate the same concept applied to a real shape.

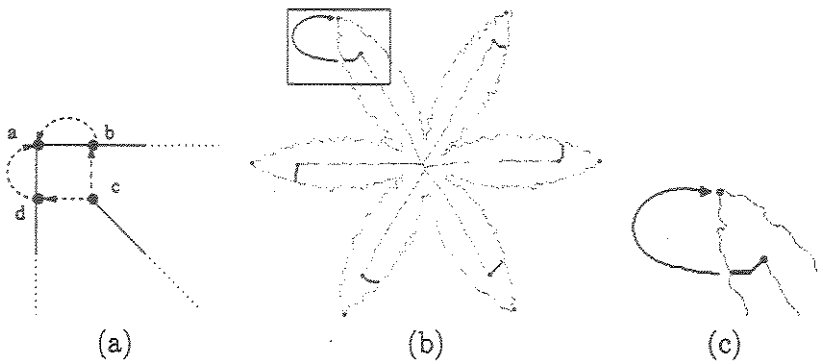


Figura 5.4: (a) Relation between skeleton and contour saliences. (b) The same concept applied to a contour. (c) A zoomed region of the figure in (b).

The correct orientation (clockwise or anti-clockwise) can be encoded in the difference image D by signaling it. Equation 5.4 must be substituted by the following algorithm applied to all pixels p in image D:

Algorithm 2:
Input: A root label map L.
Output: A signed difference image D.

1. For every pixel p of the image D, do
   - 1.1. Set $\delta_{\max} \leftarrow -\infty$.
   - 1.2. For each pixel $q \in A_4(p)$, do
     - 1.2.1. Set $\Delta \leftarrow \min\{\delta(p,q), N - \delta(p,q)\}$ and $s \leftarrow 1$.
     - 1.2.2. If $\Delta = N - \delta(p,q)$, then
       - 1.2.2.1. Set $s \leftarrow -1$.
     - 1.2.3. If $\Delta > \delta_{\max}$, then
       - 1.2.3.1. Set $\delta_{\max} \leftarrow \Delta$ and $sign \leftarrow s$.
   - 1.3. Set $D(p) \leftarrow sign \times \delta_{\max}$.

The pixels of D with absolute values greater than 5% of N are chosen to represent the internal and external skeletons. The salience points of the skeletons can be obtained by the area thresholding method described in Section 5.2. Finally, the signaled values of the skeleton salience points in D and their roots on the contour are used to locate the corresponding contour salience points, as illustrated in Figure 5.4.

Although the method works fine for convex contour points, it adds non-relevant concave points, because the external skeleton may present spurious branches due to contour rotation and scaling. Unfortunately, these non-relevant concave saliences reduce the performance of the contour salience descriptor [44, 50]. Also, if the threshold of 5% is increased to eliminate the spurious branches of the external skeleton, the method misses relevant concave points of the contour. In this chapter, the spurious branches are eliminated by an alternative skeleton labeling process and the problem is solved as follows.

The branches of the external skeleton are labeled with both, the label of their related root pixel on the contour and the length of the branch. The length-labeled skeleton image is thresholded and the resulting binary image is multiplied by the root-labeled skeleton image. These last steps remove concave contour saliences related to small branches and preserve the relevant concave saliences.

5.4 Contour Salience Descriptors

Although the salience values along the contour can not be used to locate salience points in the case of intricate and complex contours, they encode important local and global information about the contour which can be exploited to create effective shape descriptors.

An example is the descriptor based on the convex contour saliences presented in [44, 50]. Since, the problem of estimating concave points is solved now, this chapter proposes the same contour salience descriptor including the concave points (Section 5.4.1) and a new shape salience descriptor for contour segments (Section 5.4.2).

5.4.1 Contour Saliences (CS)

After determining the salience points along the contour (Section 5.3), concave points have their salience values signed negative and the salience values of convex points remain positive. One arbitrary salience point on the contour is taken as reference and the method computes the relative position of each salience point with respect to the reference point. Thus, the signed salience values and the relative position of the points form two feature vectors of the same size, which are used in the contour salience descriptor. Figure 5.5 illustrates these feature vectors for a polygon. The contour of the polygon, its reference point, the internal and external skeletons, and the respective salience points are indicated in Figure 5.5a. The plot shown in Figure 5.5b indicates the salience values versus the relative position of the points along the contour.

Whenever two contours of the same object appear in different positions (e.g., rotations and scales), they should be represented by the same salience points. However, the point taken as reference may not be the same in both. Also, the feature vectors of distinct ob-

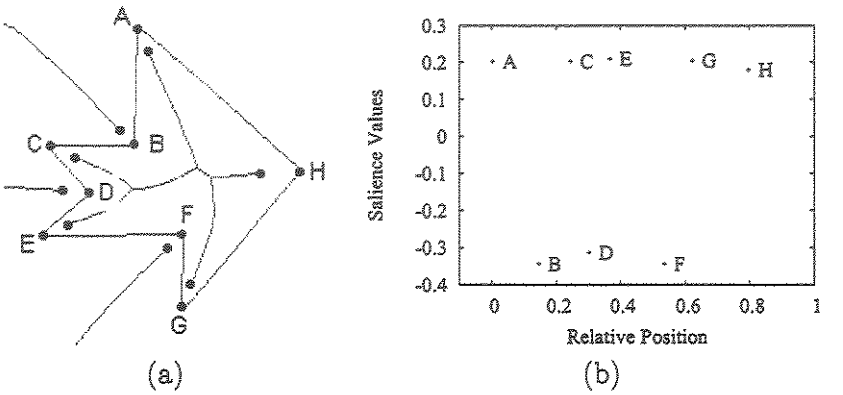


Figura 5.5: (a) Contour and skeletons of a polygon, where salience points are indicated by dots. (b) The salience values of the polygon in (a).

jects may have different sizes. Therefore, the contour salience descriptor uses a heuristic matching algorithm between contours which registers their feature vectors using the reference points and computes their similarity taking into account their difference in size. This matching algorithm is based on the algorithm proposed by Abbasi and Mokhtarian [1, 109] to match Curvature Scale Space (CSS) images, and it is described in [44, 50].

5.4.2 Segment Saliences (SS)

The segment salience descriptor is a variation of the contour salience descriptor which incorporates two improvements: the *salience values* of contour segments, in the place of salience values of isolated points, and another matching algorithm that replaces the heuristic matching by an optimum approach.

The salience values along the contour are computed as described in Section 5.2.1 and the contour is divided into a predefined number s of segments of the same size. The internal and external influence areas of each segment are computed by summing up the influence areas of its corresponding pixels. A contour segment is considered *convex*, when its accumulated external area is greater than its accumulated internal area, and it is *concave* otherwise. The *difference* between them is defined as the salience value of the contour segment, which is positive when it is convex, and negative when it is concave. These signed salience values form the feature vector of the segment salience descriptor. Algorithm 3 below presents the procedures to compute this feature vector for a given contour.

Algorithm 3:
Input: A contour ζ in an image I; number s of segments.

Output: A feature vector SS encoding the contour segment saliences.

1. Apply Algorithm 1 using the pixels in ζ as seeds and create a label map L as described in Section 5.3.1.
2. For each $t \in \zeta$, compute its internal ($H_{int}(t)$) and external ($H_{ext}(t)$) influence areas.
3. Split the contour ζ into a set $S = \{Seg_1, Seg_2, ..., Seg_s\}$ with s segments of the same size.
4. For each segment in S, compute its internal ($A_{int}(Seg_i)$) and external ($A_{ext}(Seg_i)$) influence areas as follows:
   4.1. $A_{int}(Seg_i) = \sum_{t \in Seg_i} H_{int}(t)$
   4.2. $A_{ext}(Seg_i) = \sum_{t \in Seg_i} H_{ext}(t)$
5. Compute the feature vector SS of size s as:
   5.1. $SS(i) = A_{ext}(Seg_i) - A_{int}(Seg_i)$, for $1 \leq i \leq s$

Figure 5.6 illustrates this feature vector for a contour, which is divided into 10 segments (Figure 5.6a). The curve shown in Figure 5.6b indicates the salience value of each segment along the contour.

The fixed number of segments per contour allows the use of the optimal correspondent subsequence (OCS) algorithm [156] to match feature vectors between contours. This matching algorithm is the same used in the Beam Angle Statistics (BAS) descriptor [7]. Feature vectors of the same size also simplify the storage and access methods of the image database.

5.5 Evaluation

The evaluation process consists of defining a shape database, an effectiveness measure and a set of shape descriptors for comparison.

5.5.1 Shape Database

The shape database is a set with one thousand and one hundred fish contours obtained from [136]. Since there is no semantic definition of relevant images (classes of contours) for this database, each group of relevant images is defined as one fish contour and 9 different

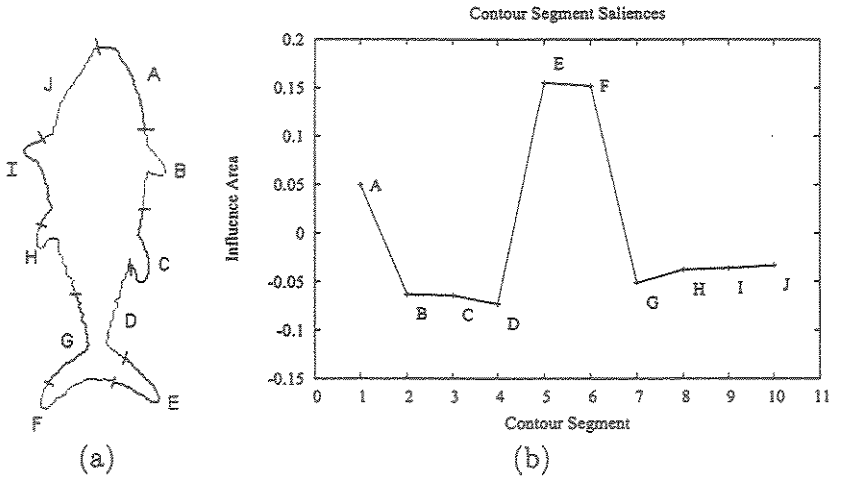


Figura 5.6: (a) A contour with 10 segments. (b) The salience values of the segments.

manifestations of rotation and scaling applied to it. Therefore, the problem consists of 1100 classes with 10 shapes each.

5.5.2 Effectiveness Measure

The experiments adopted the *query-by-example (QBE)* [8] paradigm. In the CBIR context, an image is given as an input and two types of searches are possible: *similarity range* and *similarity rank*. The search by similarity range returns the images of the database whose distance from the query image is less than a given search radius. The search by similarity rank returns a specified number of images in the increasing order of distance with respect to the query image. In both cases, the effectiveness of the system is related to the relevance of the retrieved images. It is expected that the relevant images return before non-relevant images in the second case and the non-relevant images do not return in the first case. In some applications, the relevance of the retrieved images depends on the user's opinion. However, there are several other applications where predefined classes determine groups of relevant images independent of user. Any query image in a given class should return the images of the database belonging to this class first. In such a case, it makes sense to compare descriptors based on objective measures.

The experiments of this chapter evaluate the ability of shape descriptors to distinguish between different fish contours and to identify a fish contour independent of possible rotation and scaling transformations. Note that the effectiveness of the shape descriptors apply for image retrieval and image analysis, considering the resemblance between both

Classes	Descriptor 1
class 1	$\{(1.50, 2.50), (1.50, 2.00), (2.00, 2.00), (1.00, 2.00), (1.50, 1.50)\}$
class 2	$\{(1.00, 1.00), (1.00, 2.00), (1.00, 3.00), (1.00, 4.00), (1.00, 5.00)\}$
Classes	Descriptor 2
class 1	$\{(2.00, 1.00), (2.00, 2.00), (2.00, 3.00), (2.00, 4.00), (2.00, 5.00)\}$
class 2	$\{(1.40, 1.40), (1.60, 1.40), (1.60, 1.20), (1.40, 1.20), (1.50, 1.30)\}$
Classes	Descriptor 3
class 1	$\{(1.50, 2.50), (1.50, 2.00), (1.75, 2.25), (1.25, 2.00), (1.50, 1.50)\}$
class 2	$\{(1.50, 5.50), (1.25, 5.00), (1.50, 5.00), (1.15, 5.00), (1.50, 4.50)\}$

Tabela 5.1: Coordinates of each image in classes 1 and 2 for the three hypothetical descriptors.

problems. Since each shape descriptor represents a contour as a "point" in the corresponding metric space, its effectiveness will be higher as more separate the clusters of relevant contours are in the metric space; and as more compact the clusters are in the metric space, higher will be the robustness of the shape descriptor with respect to an increase in the number of classes. Therefore, a "good" effectiveness measure should capture the concept of *separability*, and perhaps the concept of *compact-ability* for sake of robustness. More formally, the compact-ability of a descriptor indicates its invariance to the object characteristics that belong to a same class, while the separability indicates its discriminatory ability between objects that belong to distinct classes. While these concepts are commonly used to define validity measures in cluster analysis [51, 58], they seem to not have caught much attention in the literature of CBIR systems, where one of the most used effectiveness measures is *Precision* $\times$ *Recall* [112].

A simple example can be used to illustrate that *Precision* $\times$ *Recall* does not capture the separability and compact-ability concepts, and therefore, it should not be used as effectiveness measure. Consider the existence of two classes (*class 1* and *class 2*) composed by 5 images each and three different image descriptors (*descriptor 1*, *descriptor 2*, and *descriptor 3*), whose extraction algorithms create feature vectors belonging to $\mathbb{R}^2$ space. Table 5.1 shows the coordinates of each image in each class for these three hypothetical descriptors.

Figures 5.7, 5.8, and 5.9 show the classes 1 and 2 in the Cartesian plane for descriptors 1, 2 and 3, respectively.

Note that, it is reasonable to expect that the descriptor 3 will be more effective than the descriptor 2, which will be more effective than the descriptor 1. However, Figure 5.10 shows the average *Precision* $\times$ *Recall* graph for these descriptors, and even though descriptor 3 presents the best *Precision* $\times$ *Recall* curve, descriptor 1 outperforms descriptor 2.

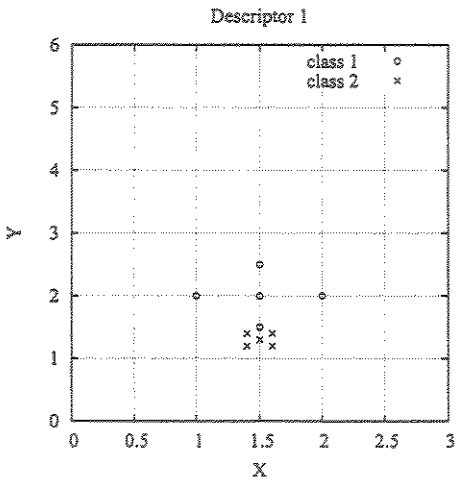


Figura 5.7: Descriptor 1

On the other hand, the concepts of separability and compact-ability seem to be better represented by the measures proposed in [44]. Figure 5.11 shows, for example, the *multiscale separability* curves for the three descriptors. Note that, descriptor 3 presents the best curve again. However, curves of descriptors 1 and 2 have the opposite behavior when compared to the *Precision* × *Recall* graph. Now, descriptor 2 is more effective than descriptor 1, as expected.

Due to these observations, the present chapter uses the concepts of compact-ability and multiscale separability proposed in [44] to evaluate the shape descriptors. The *Segment Saliences (SS)* implementation considered in this experiment used 30 segments.

5.5.3 Evaluated Descriptors

The proposed shape descriptors, contour saliences (CS) and segment saliences (SS), are compared with the following shape descriptors.

Curvature Scale Space (CSS) [1, 109]:. The CSS descriptor is used in the MPEG-7 standard and represents a multiscale organization of the curvature zero-crossing points of a planar curve. In this sense, the dimension of its feature vectors varies for different contours, thus a special matching algorithm is necessary to compare two CSS descriptors (e.g., [44]). The implementation of the CSS descriptor is a C version of the Matlab prototype presented in [108].

Beam Angle Statistics (BAS) [6, 7]: The BAS descriptor has been compared with several others [14, 30, 91, 96, 99, 109], including the CSS descriptor. In [7], it was shown that the BAS functions with 40 and 60 samples outperform all of them. The

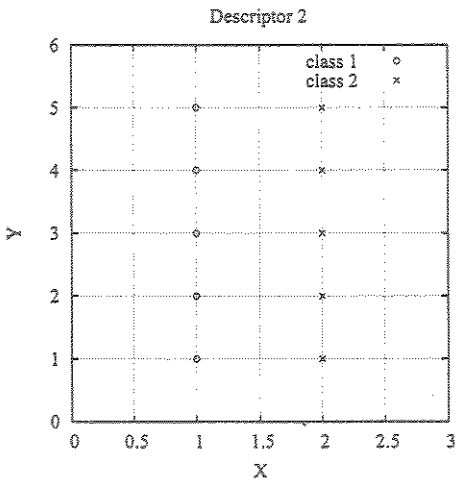


Figura 5.8: Descriptor 2

experiments of the present chapter use the BAS descriptor with 60 samples. Basically, the BAS descriptor is based on the *beams* originated from a contour pixel. A beam is defined as the set of lines connecting a contour pixel to the rest of the pixels along the contour. At each contour pixel, the angle between a pair of lines is calculated, and the shape descriptor is defined by using the third-order statistics of all the beam angles in a set of neighborhoods. The similarity between two BAS moment functions is measured by an optimal correspondent subsequence (OCS) algorithm, as shown in [7].

Convex Contour Saliences (CCS) [44, 50]: The CCS is the same descriptor described in Section 5.4.1, without the concave saliences. The CCS has outperformed Multiscale Fractal Dimension [44], Fourier Descriptors [79, 106], Moment Invariants [57, 84], CSS [1, 109] and BAS [7] with respect to the multiscale separability measure [44]. Experiments with Precision $\times$ Recall have also showed better results with the CCS as compared to CSS, Fourier Descriptors, and Moment Invariants [50]. Since the fish database is the same used in these experiments, only BAS and CSS were maintained for comparison.

Table 5.2 summarizes the set of evaluated shape descriptors.

5.5.4 Experimental Results

Figure 5.12 shows the separability curves of the evaluated descriptors. Observe that the *Contour Saliences* (CS) presents a better separability curve than the *Convex Contour Saliences* (CCS) for search radii less than 80% of their maximum distance. This indicates that the CS descriptor encodes more information (due to the concave points) than the CCS. The most relevant result is certainly the best separability curve of the *Segment*

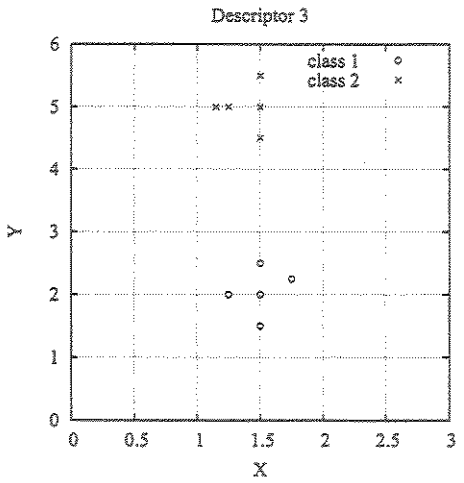


Figura 5.9: Descriptor 3.

Descriptor Id	Descriptor Name
SS	Segment Saliences
CS	Contour Saliences
CCS	Convex Contour Saliences
CSS	Curvature Scale Space
BAS	Beam Angle Statistics

Tabela 5.2: List of evaluated descriptors.

Saliences (SS) for almost all search radii.

Table 5.3 presents the compact-ability values of the evaluated shape descriptors. The higher values were found for Beam Angle Statistic (BAS) and SS, while CCS presented the lowest value. According to these experiments, the SS descriptor is more effective than the others, since it provides the best separability, and the second most robust (due to its compact-ability). This is certainly a very relevant result.

5.6 Conclusion

This chapter has presented a more robust approach to incorporate concave saliences into the contour salience descriptor and a new shape descriptor based on salience values of contour segments. They both make use of the image foresting transform as a general tool for the design of image processing operators. The results indicate segment saliences as the most effective descriptor among contour saliences, convex contour saliences [44,

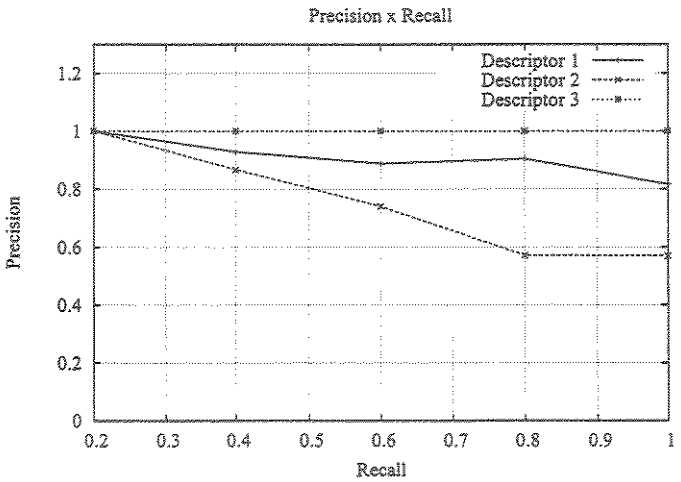


Figura 5.10: Precision vs Recall: as higher is the curve, as better is the descriptor.

Descriptor Id	Compact-ability
SS	0.93
CS	0.73
CCS	0.70
CSS	0.73
BAS	0.95

Tabela 5.3: Compact-ability values of the evaluated descriptors.

50], curvature scale space [1, 109], and beam angle statistics [6, 7], using a fish database with 11,000 images organized in 1,100 classes. They also confirm the improvement of incorporating concave saliences into the contour salience descriptor. It is important to notice that the segment salience descriptor does not require the location of salient points along the contour. In this sense, it is much simpler than the contour salience descriptor, which together with its high compact-ability make the results even more relevant.

The effectiveness in image retrieval was discussed with respect to the Precision×Recall measure and the multiscale separability [44] was proposed as a more appropriate effectiveness measure.

Ongoing developments consider the creation of shape descriptors, which combine the salience features with color- and texture-based descriptors, and applications in CBIR that use the proposed shape descriptors as effective indexing vectors.

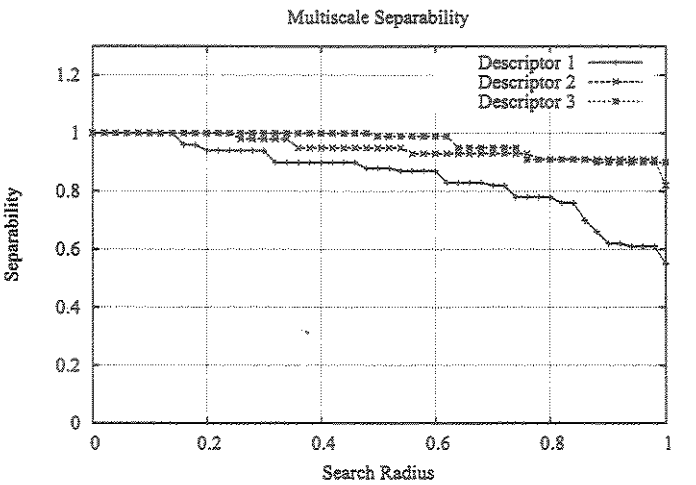


Figura 5.11: Multiscale separability: as higher is the curve, as better is the descriptor

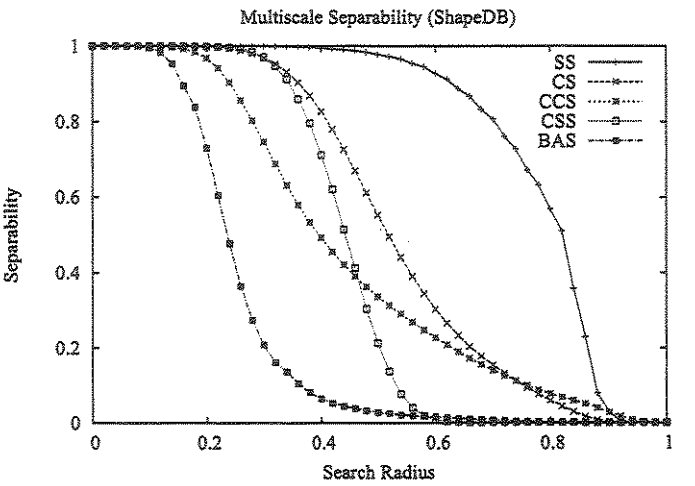


Figura 5.12: Multiscale Separability curves.

Capítulo 6

Visual Structures for Image Browsing

6.1 Introduction

Advances in data storage and image acquisition technologies have enabled the creation of large image datasets. In order to deal with these data, it is necessary to develop appropriate information systems to efficiently manage these collections. The most common retrieval approach is to attach textual metadata to each image and use traditional database query techniques to retrieve by keyword. An alternative are the so-called *Content-Based Image Retrieval (CBIR) systems*. Basically, these systems try to retrieve images similar to a user-defined specification or pattern (e.g., shape sketch, image example). Their goal is to support image retrieval based on *content* properties, e.g., shape, color or texture [139]. Research in CBIR systems is multidisciplinary and ranges from finding appropriate indexing and storage schemes for images, to cognitive problems in query specification. From the user's perspective, CBIR systems offer more flexibility in specifying queries than those based on metadata. On the other hand, they present new challenges. The first is how to *interpret* a query – e.g., when a user provides an image as input, what are the similarity criteria to be used. Another problem is *information overload* – how to present the result to the user in a meaningful way. A third issue is that of providing users with tools to *interact* with the system in order to refine their query.

Typically, the result of a query is a set of images, displayed in an Image Browser. Unfortunately, these sets are usually large, so a browsing activity must be performed. The most common result presentation technique is based on showing a two-dimensional grid of thumbnail (miniature) image versions [71, 117]. The grid is organized according to the similarity of each returned image with the query pattern (e.g., from left to right, from top to bottom). It is a $n \times m$ matrix, where position $(1, 1)$ is occupied by a thumbnail

of the query pattern, position $(1,2)$ by the one most similar to it, and so on. This helps browsing, allowing users to simply scan the grid image set as if they were reading a text [128]. This approach, however, displays retrieved images of different similarity degree at the same physical distance from the image query: e.g., images $(1,2)$ and $(2,1)$ are displayed at the same physical distance from the query pattern, but the former is more similar to it than the latter. Other display approaches try to consider relative similarity not only between the query pattern and each retrieved image, but also among all retrieved images themselves [133, 143]. These initiatives have the drawback that visually similar images which are placed next to each other can sometimes appear to merge or overlap, making them less eye-catching than if they were separated [128].

This chapter presents a new approach to these user interaction problems. This approach is based on adopting recent findings in Information Visualization techniques to provide users with semantically meaningful result presentations, and new kinds of interaction mechanisms. Information Visualization is an important field within the domain of *Human-Computer Interface (HCI)* that aims at studying the use of computer-supported, interactive, and visual representations of abstract data to amplify cognition [22, 28, 137].

The main contributions of this chapter are the following:

- presentation of two visualization techniques based on *spiral* and *concentric rings* for exploring query results in an image database. These techniques provide users new means of ranking similar images while at the same time avoid image overlap;
- description of a CBIR prototype developed which incorporates these visualization paradigms. The visualization and interaction properties of the prototype are based on the reference model described in [22].

The rest of this chapter is organized as follows. Section 6.2 characterizes the content-based image retrieval process. Section 6.3 describes the proposed visualization techniques. Section 6.4 presents implementation details. Section 6.5 discusses related work. Section 6.6 presents conclusions and ongoing work.

6.2 Content-Based Image Retrieval Systems

CBIR is centered on the notion of image *similarity* – given an image database with a large number of images, a user wants to retrieve the set of images which are most "similar" to a query pattern (usually an image). Similarity computation relies on the notion of image *descriptors*. Descriptors are defined as feature vectors whose fields contain values that encode characteristics of an image – e.g., color or texture properties. Similarity between two images is computed by measuring the distance between their feature vectors, using

specific distance functions. Usually, the degree of similarity of an image is defined as an inverse function of the distance metric, that is, the larger the distance value, the less similar the image is.

Usually, two kinds of queries are supported by CBIR systems [31]. In a *K-nearest neighbor query (KNNQ)*, the user specifies the number k of images to be retrieved closest to the query pattern. In a *range query (RQ)*, the user defines a search radius r and wants to retrieve all database images whose distance to the query pattern is less than r.

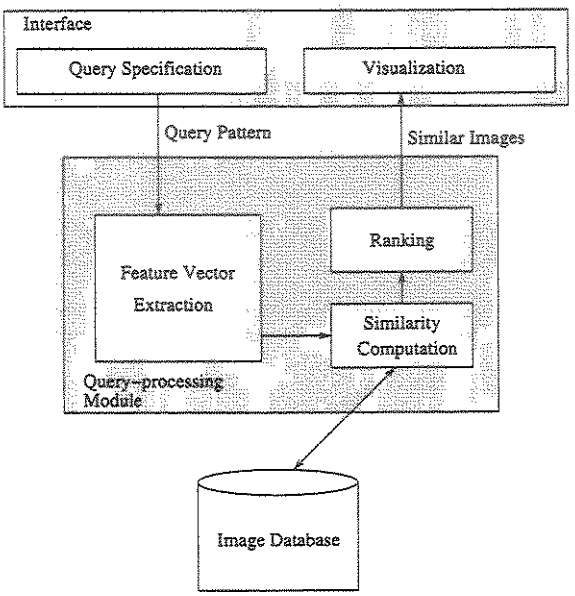


Figura 6.1: Typical image database retrieval system.

Figure 6.1 shows an overview of an image database retrieval system. The interface allows a user to specify a query by means of a query pattern and to visualize the retrieved similar images. The query-processing module extracts a feature vector from a query pattern and applies a metric distance (such as the euclidean distance) to evaluate the similarity between the query image and the database images. Next, it ranks the database images according to their similarity and forwards the most similar ones to the interface module. Database images are often indexed according to their feature vectors using structures such as the M-tree [31] to speed up retrieval and distance computation.

This chapter focuses on the interface layer. It uses Information Visualization techniques to enhance similarity comprehension and user interaction in a CBIR system.

6.3 Visual Structures Proposed

Information Visualization is a very important area in HCI. It can amplify cognition in many ways such as: increasing the memory and processing resources available to the users; reducing the search for information, e. g. due to compacting, grouping or visually relating information; enhancing the detection of patterns; enabling perceptual inference operations; using perceptual attention mechanisms for the monitoring of a large number of potential events; and encoding information in a manipulable medium [22].

One of the traditional approaches to present retrieved images in a CBIR system is based on a tabular (grid) disposition. As mentioned in the introduction, this placement affects similarity comprehension, since it displays images with different similarity degrees at the same physical distance to the query pattern.

A solution to overcome this ambiguity is to borrow techniques from the Information Visualization domain. The method proposed here is based on: (1) placing the query pattern at the center of the display, and (2) surrounding it with similar images, with physical distances and sizes proportional to their respective similarity degrees. The less similar an image is, the smaller and the farther apart from the center.

This kind of presentation relies on the fact that the user focus resides on the query pattern and the most similar images. This so-called *focus + context* approach is used to both center the user attention on the result and give the user a contextual notion of the less similar images. Besides, this approach avoids image overlapping, a common problem of some CBIR systems. Two visual structures based on this method place the images along a spiral or concentric rings.

6.3.1 Concentric Rings Presentation

A ring can be defined as a circle. In polar coordinates a circle can be expressed as $r = k$, where r is the radial distance, and k is a constant. Successive rings are built by changing the k value. Moreover, all rings have the same center and successive rings become closer as k increases. The rings are filled from the innermost ring to the outermost one, according to image ranking. Figure 6.2a illustrates the concentric ring visual structure implemented, where dots represent image thumbprints.

6.3.2 Spiral Presentation

The most common planar spirals are the spiral of Archimedes, the hyperbolic spiral and the logarithmic spiral. In order to contemplate the characteristics proposed into our method, a spiral line should become closer to itself as it loops away. This aspect is directly related to the images' size variation along the structure. Since hyperbolic and

logarithmic spirals move rapidly away from the origin, they are not appropriate to our goal. Thus, the choice was Archimedes's spiral, expressed in a polar equation as $r = k\theta^a$, where r is the radial distance, θ is the polar angle, k is a constant and a is a constant which determines how tightly the spiral "wraps" around itself. Figure 6.2b illustrates the spiral adopted, considering $k = 2.5$ and $a = -1.5$. Observe that the greater θ is, the tighter the spiral line becomes, enforcing the focus on the central region.

There are two different ways to display images along a spiral line. The first associates the image ranking to the spiral line, in such a way that the images are disposed successively, at regular distances (Figure 6.2b). This approach, however, does not present the real similarity degree. A second alternative maps the similarity degree to the spiral line, that is, the image distance to the query pattern is proportional to its similarity degree (Figure 6.2c).

6.4 Implementation

This section describes the prototype implemented. It is written in Builder C++, running on Windows. It was tested on a database of 11000 fish images and uses two shape descriptors called *Multiscale Fractal Dimension* and *Shape Saliences* [42, 43, 50]. This is part of a biodiversity information system, where users (biologists) explore a database containing images and textual descriptions to find out details about species. Details of this project are outside the scope of the chapter [44].

6.4.1 Formalizing the Visualization Framework

Research in Information Visualization often uses the reference model of [22] as a basis to study the cognitive enhancement provided by visual representations. This model defines a way to analyze successive transformations from raw data to visual representations, taking into account possible human interactions within this process through three transformation stages: *data transformation (DT)*, *visual mapping (VM)* and *view transformation (VT)*. The first stage considers that raw data (data in some idiosyncratic format) are initially transformed into data tables (DT). Tables are next mapped to structures with graphical properties – visual structures, displayed on a screen (VM). Finally, these static structures are transformed into views, which are dynamic, interactive and information-enriched representations (VT).

Many techniques have been proposed to deal with each transformation step and the underlying data structures [22]. *Details-on-demand* is a method used within the data transformation stage to expand a small set of objects revealing more information about it [3, 22]. *Pan* and *zoom* are commonly used together within view transformations to

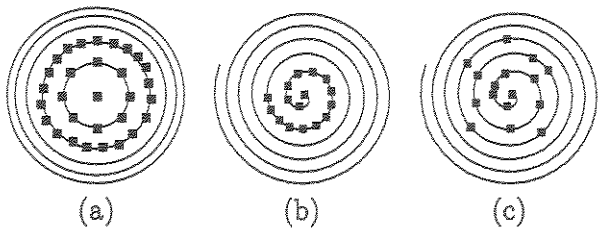


Figura 6.2: Visual Structures. (a) Concentric rings. (b) Spiral mapping image ranking. (c) Spiral encoding image similarity degree. Note that image size reduction along the structures is not shown.

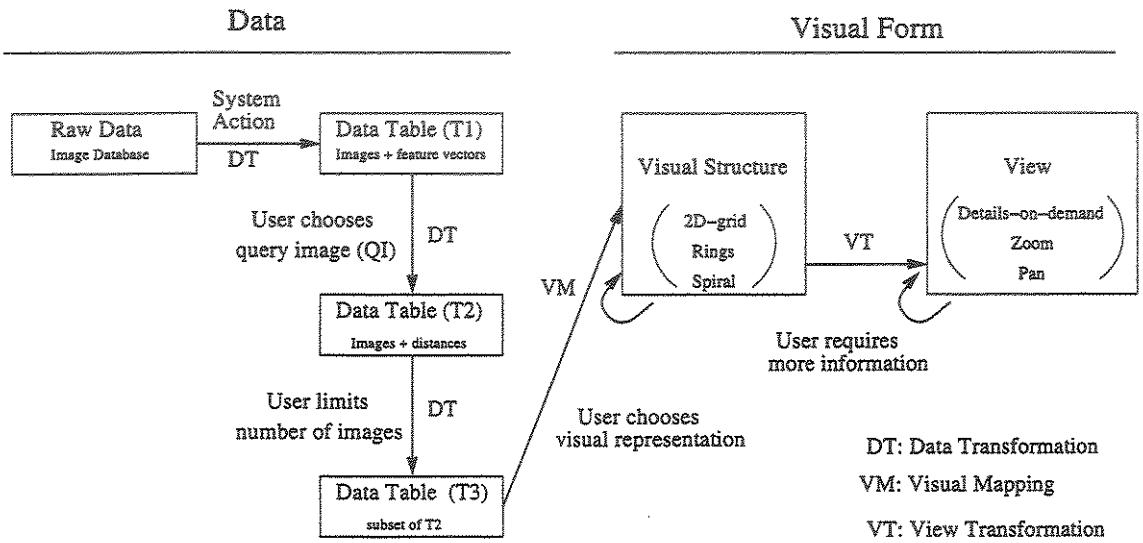
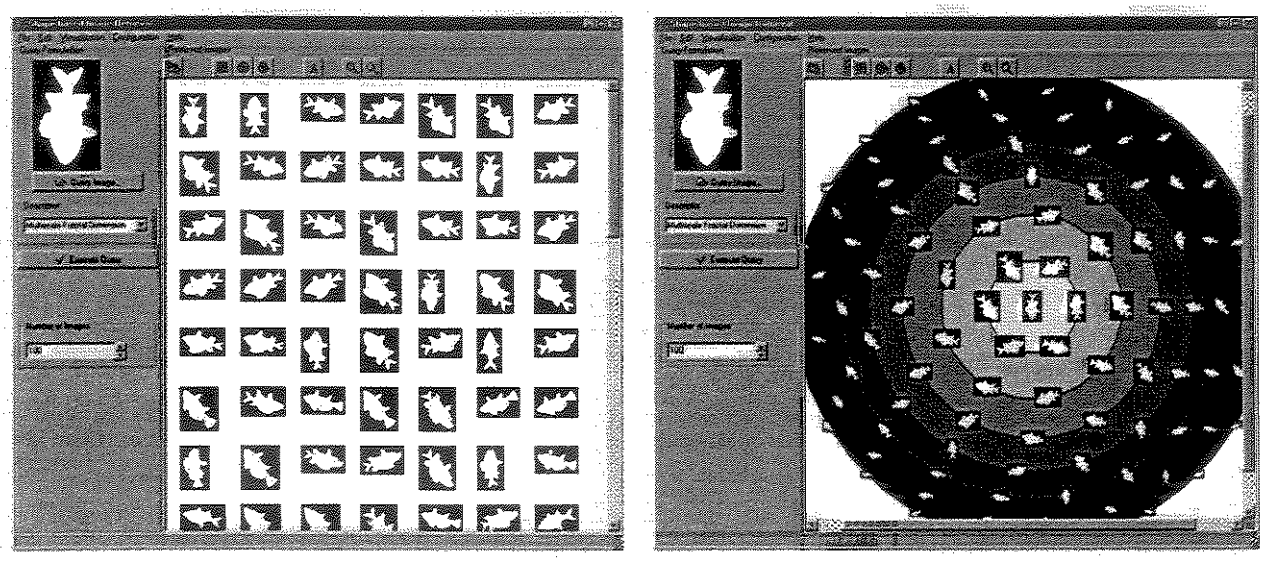


Figura 6.3: Information Visualization phases on a CBIR system.

change the viewer's position and to focus on a specific set of data. The *focus + context* approach is also used within a view transformation stage. It simultaneously combines overview (context) and detail information (focus), using distortion or other specific techniques.

Figure 6.3 analyzes the architecture of our CBIR prototype under this reference model. The image database represents the raw data. Image processing algorithms automatically extract feature vectors that encode the image content. This extraction phase is the first data transformation, and generates a data table comprised of each image and its feature vector (T1). When a user inputs a query image (QI), a second data transformation occurs: QI's feature vector (FQI) is automatically extracted, and the system executes a matching algorithm to compute the distance from the FQI to feature vectors stored in T1, thus generating a second data table (T2). This table stores the distances from FQI to the



(a) (b)

Figura 6.4: Prototype screen shots. (a) 2D grid approach. (b) Concentric rings approach.

feature vector of each database image. A third data transformation occurs when the user specifies a limit to the number of images to retrieve, leading to a third data table (T3) that is a subset of T2. Next, the user chooses the visual structure to be used: traditional (2D grid), spiral or concentric rings. All three visual structures take into account the distance values stored into T3. Finally, the user can interactively manipulate the display using details-on-demand, zoom, pan and focus + context, generating new views of the chosen visual structure and improving user cognition.

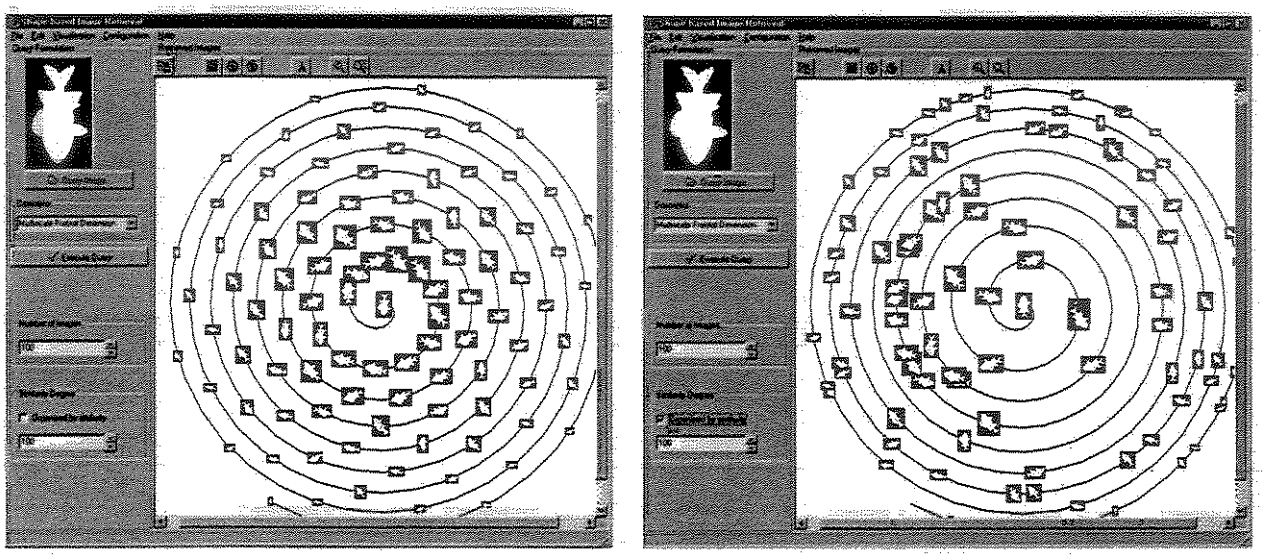
6.4.2 A Sample User Session

Consider the following sample session. Initially, a user specifies a query by providing a query image (the query pattern). Next, the user chooses the descriptor for similarity computation and the visual structure for displaying the query result – 2D grid, rings or spiral.

The 2D grid-based traditional approach just obeys the rank. When it reaches the horizontal end of the screen, it continues the sequence of images on the next line, a typical use of the so-called folding technique [22]. Figure 6.4a shows a screen copy of this standard visualization approach. The query image is on the left topmost part of the screen. The result of the query, organized in a 2D grid, is on the large part on the right. Since we use shape descriptors, the closest results are fish images whose shapes are most similar to the query pattern's shape. Thus, image rotation or scaling are not taken into

consideration in similarity computation. This is a nice property of the shape descriptors used – see [42, 43, 50] for more details.

Figure 6.4b shows the result of the same query using the *concentric rings* visualization approach. This screen shot enhances the ring structure with increasing levels of gray to help focus user attention on the center and provide a better separation among rings (another technique in visualization theory). The query image is at the center of the rings. Images at rings farther from the center are less similar than those along closer rings.



(a) (b)

Figura 6.5: Prototype screen shots. Spiral placement based on (a) image ranking and (b) similarity.

In a similar fashion, the *spiral* approach also places the query image in the center, and fills the spiral with the retrieved images. Figure 6.5 presents the two available spiral variants. Figure 6.5a shows a spiral in which images are placed successively, at regular distances. Figure 6.5b, in turn, places the ranked images within the spiral considering their similarity degree. This latter approach, however, can overlap images with similar distance to the query pattern.

Users can interact with the result in many ways. Besides zoom and pan operations, they can select a specific image as a new query, or obtain a detail-on-demand box with a real-sized image and its filename. The user can also control the number of images displayed (simulating a KNN query) for all three visual representations. In the case of spiral representation, the user can threshold the retrieved images by their degree of similarity. This corresponds to a range query, where the search radius is controlled by the user.

The user can specify a new query either by selecting an image from the retrieved image set or by providing a new image file name. Besides, the user can provide new query parameters (e.g., kind of descriptor or number of retrieved images) via the textual controls at the left part of the screen.

6.4.3 Relevance Feedback

Relevance feedback is a commonly accepted method to improve interactive retrieval effectiveness [103]. Basically, it is composed of three steps: (a) an initial search is made by the system with a user-supplied query pattern, returning a small number of images; (b) the user then indicates which of the retrieved images are useful (relevant); (c) finally, the system automatically reformulates the original query based upon user relevance judgments. This process can continue to iterate until the user is satisfied.

The proposed visual structures can also be used to improve user interaction in the relevance feedback process. Two kinds of interaction based on *direct manipulation* are foreseen. First, a user can move images along the spiral line. By taking an image away from the spiral center, the user informs the system that this image is not relevant. The opposite situation is also true: moving an image closer to the spiral center increases its relevance for future queries. A similar interaction can occur on the concentric ring visual structure. In this case, users can move an image across rings – relevance increases as the image is moved to a position closer to the center.

6.4.4 Experimentation

Experiments conducted so far did not gather enough data to prove the superiority of spiral or concentric rings over 2D grids. So far, our experiments have been conducted with a limited number of users, that are not experts on research on fish. This has been a limitation factor on interface evaluation, since we would have to consider many kinds of user profile. Nevertheless our experiments allow the following preliminary conclusions:

- alternative (multiple) views of a result are much more useful than just the usual 2D grid, offering users distinctive perceptions of relative distances and similarities;
- when the query for k nearest neighbor images involves large values of k, the result clutters the screen. In this case, spiral and ring presentations help zooming into the desired result. For small values of k (typically when results can be seen in one horizontal line) users see no advantage in using alternatives to 2D grid.
- users were not aware that extended visualization presentations were possible. Faced with alternatives, they began demanding further extensions. The prototype presents

the results ranking all images w.r.t. the query pattern. An extension would be to allow clustering images according to the relative distance to each other. Another request is for 3D presentation, though recent results seem to indicate that for this kind of query 2D presentations are better cognition-wise [35].

6.5 Related Work

Information Visualization is attracting considerable attention in several domains, such as data mining, data exploration and knowledge discovery (e.g., [90, 155]). In particular, classification in data mining is often visualized in terms of data clusters, where each class instance is presented as a point in a 2D or 3D space. Each cluster represents a data class, and the distances among clusters allow users to deduce the relative similarity among classes and their instances.

2D grid presentation can be found in several image database systems [71, 117, 139]. [12] and [35] try to improve this visual structure by studying zoom properties to enhance image browsing. Rodden *et al.* [128], in turn, investigates whether it benefits users to have sets of thumbnails arranged according to their similarity, so images that are alike are placed together. They describe experiments to examine whether similarity-based arrangements of the candidate images help in picture selection.

Stan *et al.* [143] describe an exploration system for an image database, which deals with a tool for visualization of the database at different levels of details based on a multi-dimensional scaling technique. This visualization technique groups together perceptual similar images in a hierarchy of image clusters. Retrieved images can overlap. The overlap problem is also found in the El Niño image database [133]. In this context, Tian *et al.* [150] propose a PCA (Principal Component Analysis)-based image browser which looks into an optimization strategy to adjust the position and size of images in order to minimize overlap (maximize visibility) while maintaining fidelity to the original positions which are indicative of mutual similarities.

Spirals and rings are used to visualize information in different domains [23, 87, 157, 161]. [23] and [157] investigate the use of spirals to visualize time-series. They display data along a spiral to highlight serial attributes along the spiral axis and periodic ones along the radii. Mackinlay *et. al.* [87], in turn, use a spiral for calendar visualization, building iconic representations of past daily calendar entries, positioned on a spiral. A radial layout is used in [161] to visualize graphs. In this approach, graph nodes are arranged on concentric rings around the focus node. Each node lies on the ring corresponding to its shortest network distance from the focus.

6.6 Conclusion

This chapter presented a new approach to improve user interaction in CBIR systems based on applying Information Visualization research to construct CBIR interfaces. It discusses two visualization techniques based on Spiral and Concentric Rings to explore query results. These visual structures are centered on keeping user focus on the query image and on the most similar retrieved images. These strategies improve traditional 2D grid presentation and avoid image overlaps, commonly found in CBIR systems.

Ongoing work includes the finalization of user experiments, and the definition of a new visualization strategy. This strategy extends the proposed methods by considering the mutual similarities among retrieved images. At the same time, relevance feedback principles are being incorporated to the prototype.

Capítulo 7

Conclusões

7.1 Contribuições

Esta tese desenvolveu pesquisa combinando aspectos de processamento de imagens, bancos de dados e bibliotecas digitais. Ela abordou os desafios apresentados por sistemas de informação ambiental mencionados no Capítulo 1, a saber: a necessidade de mecanismos de *interação* com usuário para facilitar especificação de consultas; a dificuldade em combinar *mecanismos de consulta* por conteúdo a bancos de imagens e consulta a bancos de dados geográficos; e a complexidade do *gerenciamento* diferenciado de dados de natureza tão distinta. Esta combinação de enfoques no processamento de uma consulta permitiu criar um novo tipo de sistema para apoio a pesquisas de cientistas em biodiversidade.

O resultado final foi a especificação e implementação parcial de um protótipo de ambiente computacional que combina o gerenciamento de imagens de seres vivos e dados espaciais para aplicações ambientais de biodiversidade. Este ambiente permite consultas cujos predicados componham recuperação baseada em conteúdo, de bancos de imagens, com recuperação baseada em localização e características geográficas e ecológicas. O trabalho foi implementado com base em dois conceitos: o uso de componentes de software e a adoção de padrões e mecanismos de bibliotecas digitais.

As principais contribuições deste trabalho são:

1. Especificação de um ambiente para gerenciamento integrado de fotos, mapas e metadados usando o arcabouço de Bibliotecas Digitais, que permite a cientistas a extração de informação segundo suas necessidades;

2. Criação de um componente de busca para recuperação de imagens por conteúdo;

3. Proposta de novos descritores de imagens para suporte à recuperação de imagens segundo o seu conteúdo (forma dos objetos). Testes envolveram a avaliação de 50

novos descritores de forma, definidos através do uso de diferentes implementações dos algoritmos de extração e comparação de vetores de características. Para cada descritor, 1100 consultas foram realizadas (55000 no total) numa base de dados com 11000 formas de peixes. Apenas os melhores descritores, avaliados através das métricas de compactabilidade e separabilidade, são apresentados nos Capítulos 4 e 5;

4. Proposta de novas estruturas visuais para visualização de resultados de consultas em sistemas de recuperação de imagens por conteúdo;

5. Implementação e validação parcial da arquitetura proposta para um tipo específico de imagem (peixes) e perfil de usuário (biólogos) em uma aplicação real de suporte ao processo de identificação de espécies de peixe. Esta implementação está sendo usada no Department of Fisheries and Sciences na Virginia Tech [45], para apoio ao ensino de biólogos na área de ictiologia.

7.2 Extensões

Há várias extensões previstas, tanto do ponto de vista teórico quanto de implementação, para as diferentes camadas da arquitetura. Algumas destas extensões, inclusive, já estão sendo analisadas. Os trabalhos futuros incluem:

- **Projeto de Novas Interfaces:** Este trabalho de pesquisa teve como objetivo o suporte a consultas que combinam diferentes fontes de evidência: descritores textuais, dados espaciais e descritores de conteúdo de imagens. O Capítulo 6 apresentou novas estruturas visuais para visualização de resultados de consultas em bancos de dados de imagens. Este trabalho pode ser estendido, com a inclusão de estruturas visuais intuitivas para suporte na especificação de consultas e visualização de resultados que envolvam as diferentes evidências consideradas. O projeto destas interfaces deve considerar a realização de consultas em mapas, seleção de descritores e imagens de consulta e o uso de palavras-chave.

- **Otimização de Consultas:** O trabalho não abordou aspectos de desempenho no processamento das consultas. A definição e implementação de regras para otimização de consultas que envolvam as diferentes evidências precisam ser investigadas. Propostas de solução para este problema incluem o uso de linguagens de consulta e novas álgebras baseadas em similaridade recentemente propostas [2, 9, 10, 27, 98, 114, 142].

- **Combinação de Descritores:** Nas aplicações ambientais consideradas, o processamento de consultas envolvendo o conteúdo de imagens pode envolver diferentes

propriedades, tais como cor, textura e forma. Uma extensão neste sentido consiste em investigar a utilização de mecanismos para combinação de descritores codificando estas diferentes propriedades. As abordagens mais comuns usam a atribuição de diferentes pesos para codificar a importância de cada propriedade [71]. Outras abordagens, baseadas em técnicas de fusão de consultas [34], também podem ser consideradas.

- **Uso de Ontologias:** A arquitetura proposta pode ser estendida para suporte ao uso de ontologias na definição e processamento de consultas. Ontologias podem ser utilizadas no gerenciamento das diferentes fontes de evidências: metadados de imagens podem estar associados a um conjunto de termos definidos em uma ontologia [113, 134]; ontologias específicas para as aplicações ambientais podem guiar consultas envolvendo, por exemplo, informação sobre ecossistema ou habitat de uma espécie. Ainda, ontologias espaciais podem ser utilizadas na conversão de nomes de lugares definidos em consultas textuais em coordenadas na superfície terrestre.

- **Extensão para outros Domínios:** O trabalho de implementação foi centrado em sistemas de biodiversidade, considerando as necessidades deste tipo de usuário e especificidade dos dados utilizados, em especial informação sobre coletas e os metadados. A arquitetura, no entanto, é genérica. Assim, outra extensão possível seria testar a implementação para outros tipos de sistema ambiental – por exemplo, em modelagem de ecossistemas ou em estudos de impacto ambiental. Neste caso, seria necessário um novo tipo de análise de conteúdo de imagens. Além disso, haveria necessidade de prover outras combinações de consulta, pois os dados usados nestes sistemas são bem diferentes daqueles típicos de sistemas de biodiversidade.

Referências Bibliográficas

[1] S. Abbasi, F. Mokhtarian, and J. Kittler. Enhancing CSS-based Shape Retrieval for Objects with Shallow Concavities. *Image and Vision Computing*, 18(3):199–211, February 2000.

[2] S. Adali, P. A. Bonatti, M. L. Sapino, and V. S. Subrahmanian. A Multi-Similarity Algebra. In *Proceedings ACM SIGMOD International Conference*, pages 402–413, Seattle, Washington, USA, June 2-4 1998.

[3] C. Ahlberg and B. Snheiderman. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 313–317, 479–480, Vienna, Austria, 1994.

[4] L. A. N. Amaral, P. C. Ivanov, N. Aoyagi, I. Hidaka, S. Tomono, A. L. Goldberger, H. E. Stanley, and Y. Yamamoto. Behavioral-Independent Features of Complex Heartbeat Dynamics. *Physical Review Letters*, 86(26):6026–6029, 2001.

[5] S. Analyst, 2004. http://www.speciesanalyst.net (as of October 2004).

[6] N. Arica and F. T. Y. Vural. A Perceptual Shape Descriptor. In *Proceedings of the International Conference on Pattern Recognition*, pages 375–378, Madison, Wisconsin, USA, 2002.

[7] N. Arica and F. T. Y. Vural. BAS: A Perceptual Shape Descriptor Based on the Beam Angle Statistics. *Pattern Recognition Letters*, 24(9-10):1627–1639, June 2003.

[8] Y. A. Aslandogan and C. T. Yu. Techniques and Systems for Image and Video Retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):56–63, January/February 1999.

[9] S. Atnafu, L. Brunie, and H. Kosch. Similarity-Based Operators and Query Optimization for Multimedia Database Systems. In *Proceedings of the Interna-*

tional Database Engineering and Application Symposium, pages 346–355, Grenoble, France, 2001.

[10] S. Atnafu, L. Brunie, and H. Kosch. Similarity-Based Operators in Image Database Systems. In *Proceedings of the Advances in Web-Age Information Management, Second International Conference*, pages 14–25, Xi'an, China, 2001.

[11] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, and C.-F. S. R. Jain. Virage Image Search Engine: An Open Framework for Image Management. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 76–87, Bellingham, WA, USA, 1996.

[12] B. B. Bederson. Photomesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 71–80, Orlando, FL, USA, 2001.

[13] R. Bellman. On a Routing Problem. *Quarterly of Applied Mathematics*, 16:87–90, 1958.

[14] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24):509–522, 2002.

[15] BioGIS. BioGIS - Israel Biodiversity Information System. http://www.eti.uva.nl/ (as of October 2004).

[16] BIOTA/FAPESP. SinBiota (Sao Paulo State's Biodiversity Information System). http ://www.biota.org.br/sia (as of October 2004).

[17] H. Blum. Biological Shape and Visual Sciences (Part I). *Journal of Theoretical Biology*, 38:205–287, 1973.

[18] M. Bober. MPEG-7 Visual Shape Descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):716–719, June 2001.

[19] R. Brunelli and O. Mich. On the Use of Histograms for Image Retrieval. In *Proceedings IEEE International Conference on Multimedia Computing and Systems*, pages 143–147, Florence, Italy, June 1999.

[20] G. Camara, R. Thome, U. Freitas, and A. V. Monteiro. Interoperability in Practice: Problems in Semantic Conversion from Current Technology to OpenGIS. www.dpi.inpe.br/gilberto/papers/interop99.pdf (July 2004).

[21] M. Cammert, C. Heinz, J. Kramer, M. Schneider, and B. Seeger. A Status Report on XXL - A Software Infrastructure for Efficient Query Processing. *Data Engineering Bulletin*, 26(2):12–18, 2003.

[22] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 1999.

[23] J. V. Carlis and J. A. Konstan. Interactive Visualization of Serial Periodic Data. In *Proceedings of the 11th Annual Symposium on User Interface Software and Technology*, San Francisco, CA, USA, November 1998.

[24] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image Segmentation Using Expectation-Maximization and its Application to Image Querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, August 2002.

[25] V. Castelli and L. D. Bergman, editors. *Image Databses. Search and Retrieval of Digital Imagery*. John Wiley Sons, New York, NY, USA, 2002.

[26] S. F. Chang, T. Sikora, and A. Puri. Overview of the MPEG-7 Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):688–695, June 2001.

[27] S. Chaudhuri and L. Gravano. Optimizing Queries over Multimedia Repositories. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 91–102, Montreal, Quebec, 1996.

[28] E. H. Chi. A Taxonomy of Visualization Techniques Using the Data State Reference Model. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 69–76, Salt Lake City, Utah, USA, 2000.

[29] M. G. Christel, A. M. Olligschlaeger, and C. Huang. Interactive Maps for a Digital Video Library. *IEEE Multimedia*, 7(1):60–67, 2000.

[30] G. Chuang and C.-C. Kuo. Wavelet Descriptor of Planar Curves: Theory and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(1):56–70, 1996.

[31] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In *Proceedings of 23rd International Conference on Very Large Data Bases*, pages 426–435, Athens, Greece, 1997.

[32] CITIDEL. Computing and Information Technology Interactive Digital Educational Library, 2003. http://www.citidel.org (as of October 2004).

[33] CMI. Conservation Management Institute, http://fwie.fw.vt.edu/indexnew.html (as of October 2004).

[34] T. A. S. Coelho, P. P. Calado, L. V. Souza, B. Ribeiro-Neto, and R. Muntz. Image Retrieval Using Multiple Evidence Ranking. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):1–10, July/August 2003.

[35] T. T. A. Combs and B. B. Bederson. Does Zooming Improve Image Browsing? In *Proceedings of the Fourth ACM International Conference on Digital Libraries*, pages 130–137, Berkeley, California, USA, 1999.

[36] L. Costa, A. Campos, and E. Manoel. An Integrated Approach to Shape Analysis: Results and Perspectives. In *Proceedings of the International Conference on Quality Control by Artificial Vision*, pages 23–34, Le Creusot, France, May 2001.

[37] L. Costa and R. C. Jr. *Shape Analysis and Classification: Theory and Practice*. CRC Press, Boca Raton, FL, USA, 2001.

[38] L. da F. Costa and L. F. Estrozi. Multiresolution Shape Representation without Border Shifting. *Electronic Letters*, 35(21):1829–1830, 1999.

[39] L. da F. Costa and E. T. M. Manoel. A Shape Analysis Framework for Neuromorphometry. *Network*, 13:283–310, 2002.

[40] R. da S. Torres, C. G. da Silva, C. B. Medeiros, and H. V. da Rocha. Visual Structures for Image Browsing. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pages 167–174, New Orleans, Lousiana, USA, November 2003. ACM Press.

[41] R. da S. Torres and A. X. Falcão. Contour Salience Descriptors for Effective Image Retrieval and Analysis. *Image and Vision Computing*, 2004. Submitted.

[42] R. da S. Torres, A. X. Falcão, and L. da F. Costa. Shape Description by Image Foresting Transform. In *Proceedings of the 14th International Conference on Digital Signal Processing*, volume 2, pages 1089–1092, Santorini, Greece, July 2002.

[43] R. da S. Torres, A. X. Falcão, and L. da F. Costa. A Graph-based Approach for Multiscale Shape Analysis. Technical Report IC-0303, Institute of Computing, University of Campinas, Campinas, SP, Brazil, January 2003.

[44] R. da S. Torres, A. X. Falcão, and L. da F. Costa. A Graph-based Approach for Multiscale Shape Analysis. *Pattern Recognition*, 37(6):1163–1174, June 2004.

[45] R. da S. Torres, E. Halleman, R. E. Jenkins, and N. M. Burkhead. EKEY, http://virginia.cc.vt.edu:8086/ekey (as of October 2004).

[46] R. da S. Torres, C. B. Medeiros, R. Q. Dividino, M. A. Figueiredo, M. A. Goncalves, E. A. Fox, and R. Richardson. Using Digital Library Components for Biodiversity Systems. In *Proceedings of the 4th Joint ACM/IEEE Conference on Digital Libraries*, page 408, Tucson, AZ, June 2004.

[47] R. da S. Torres, C. B. Medeiros, M. A. Goncalves, and E. A. Fox. A Digital Library Framework for Biodiversity Information Systems. *International Journal on Digital Libraries*, 2004. Submitted.

[48] R. da S. Torres, C. B. Medeiros, M. A. Goncalves, and E. A. Fox. An OAI Compliant Content-Based Image Search Component. In *Proceedings of the 4th Joint ACM/IEEE Conference on Digital Libraries*, page 418, Tucson, AZ, June 2004.

[49] R. da S. Torres, C. B. Medeiros, E. M. Hallerman, M. A. Goncalves, and E. Fox. Integrating Image and Spatial Data for Biodiversity Information Management. Technical Report IC-04-12, Institute of Computing, UNICAMP, Campinas, SP, Brazil, 2004.

[50] R. da S. Torres, E. M. Picado, A. X. Falcão, and L. da F. Costa. Effective Image Retrieval by Shape Saliences. In *Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing*, pages 49–55, São Carlos, São Paulo, Brazil, 12-15 October 2003.

[51] D. Davies and D. Bouldin. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.

[52] S. de P. Matias. Query Processing in BIOTA Biodiversity Database. Master's thesis, Institute of Computing, UNICAMP, Campinas, SP, Brazil, 2000.

[53] A. del Bimbo. *Visual Information Retrieval.* Morgan Kaufmann Publishers, San Francisco, CA, USA, 1999.

[54] J. V. den Bercken, B. Blohsfeld, J.-P. Dittrich, J. Krämer, T. Schäfer, M. Schneider, and B. Seeger. XXL - A Library Approach to Supporting Efficient Implementations of Advanced Database Queries. In *Proceedings of the 27th International Conference on Very Large Data Bases*, pages 39–48, Roma, Italy, 2001.

[55] E. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1:269–271, 1959.

[56] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York, 1973.

[57] S. A. Dudani, K. J. Breeding, and R. B. McGhee. Aircraft Identification by Moment Invariants. *IEEE Transactions on Computers*, 26(1):39–45, January 1977.

[58] J. Dunn. Well Separated Clusters and Optimal Fuzzy Partitions. *Cybernetics*, 4:95–104, 1974.

[59] ESSEX, 2004. http://br.endernet.org/~akrowne/elaine/essex/index.html (as of October 2004).

[60] ETANA. Managing Complex Information Applications: An Archaeology Digital Library, 2004. http://feathers.dlib.vt.edu (as of October 2004).

[61] ETI. World Biodiversity Database, 2002. http://www.eti.uva.nl/Database/WBD.html (as of October 2004).

[62] ETI. Expert Center for Taxonomic Identification GCMD - The Linnaeus System, 2004. http://www.biogis.huji.ac.il/ (as of October 2004).

[63] A. Falcão and B. S. da Cunha. Multiscale Shape Representation by Image Foresting Transform. In M. Sonka and K. Hanson, editors, *Proceedings of SPIE on Medical Imaging*, volume 4322, pages 1091–1100, San Diego, CA, Feb 2001.

[64] A. Falcão, B. S. da Cunha, and R. A. Lotufo. Design of Connected Operators Using the Image Foresting Transform. In M. Sonka and K. Hanson, editors, *Proceedings of SPIE on Medical Imaging*, volume 4322, pages 468–479, San Diego, CA, Feb 2001.

[65] A. Falcão, L. da F. Costa, and B. da Cunha. Multiscale Skeletons by Image Foresting Transform and its Applications to Neuromorphometry. *Pattern Recognition*, 35(7):1571–1582, April 2002.

[66] A. Falcão, L. da F. Costa, and B. da Cunha. Erratum to Multiscale Skeletons by Image Foresting Transform and its Applications to Neuromorphometry: [Pattern Recognition 35(7) (2002) 1571-1582]. *Pattern Recognition*, 36(12):3013, Dec 2003.

[67] A. X. Falcão, J. Stolfi, and R. A. Lotufo. The Image Foresting Transform: Theory, Algorithms, and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):19–29, Jan 2004.

[68] G. Faria. A Spatio-Temporal Database for Development of Applications in Geographic Information Systems. Master's thesis, IC - UNICAMP, 1998. Advisor: Claudia Bauzer Medeiros.

[69] R. Fileto. Issues on Interoperability and Integration of Heterogeneous Geographical Data. In *Proceedings of the Third Brazilian Symposium on Geoinformatics*, pages 129–136, 2001.

[70] FishBase. www.fishbase.org (as of October 2003).

[71] M. Flickner, H. Sawhney, W. Niblack, Q. H. J. Ashley, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by Image and Video Content: the QBIC System. *IEEE Computer*, 28(9):23–32, Sep 1995.

[72] A. Frieze. Minimum Paths in Directed Graphs. *Operational Research Quarterly*, 28(2,i):339–346, 1977.

[73] GBIF. Global Biodiversity Information Facility (GBIF). http://www.gbif.org (as of October 2004).

[74] GeoServer. http://geoserver.sourceforge.net (as of October 2004).

[75] GML. Geography Markup Language (GML). http://opengis.net/gml (as of October 2004).

[76] M. A. Goncalves and E. A. Fox. 5SL: a language for declarative specification and generation of digital libraries. In *Proceeding of the Second ACM/IEEE-CS Joint Conference on Digital libraries*, pages 263–272, Portland, Oregon, USA, 2002. ACM Press.

[77] M. A. Goncalves, R. K. France, and E. A. Fox. MARIAN: Flexible Interoperability for Federated Digital Libraries. In *Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries*, pages 173–186, Germany, 2001.

[78] M. A. Goncalves, P. Marther, J. Wang, Y. Zhou, M. Luo, R. Richardson, R. Shen, L. Xu, and E. A. Fox. Java MARIAN: From an OPAC to a Modern Digital Library System. In *String Processing and Information Retrieval: 9th International Symposium, SPIRE 2002*, pages 194–209, Lisbon, Portugal, September 2002.

[79] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, Reading, MA, USA, 1992.

[80] R. H. Guting. An Introduction to Spatial Database Systems. *The International Journal on Very Large Data Bases*, 4(3):357–399, October 1994.

[81] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernatics*, 3(6):610–621, 1973.

[82] L. Helfrich, T. Newcomb, E. Halleman, and K. Stein. EFISH, http://www.cnr.edu/efish (as of October 2004).

[83] J. S. Hong, H. Chen, and J. Hsiang. A Digital Museum of Taiwanese Butterflies. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, pages 260–261, San Antonio, Texas, United States, 2000. ACM Press.

[84] M. K. Hu. Visual Pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962.

[85] Interop. International Conference and Workshop on Interoperating Geographic Information Systems. Web site http://www.ncgia.ucsb.edu/conf/interop97, 12 1997. Site address valid as of 07/99.

[86] IUBS, CODATA, and IUMS. Species 2000. http://www.sp2000.org (as of October 2004).

[87] G. G. R. J. D. Mackinlay and R. DeLine. Developing Calendar Visualizers for the Information Visualizer. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology*, pages 109–118, Marina del Rey, California, USA, 1994.

[88] G. Janee and J. Frew. The ADEPT Digital Library Architecture. In *Proceeding of the Second ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 342–350, Portland, Oregon, USA, 2002. ACM Press.

[89] R. E. Jenkins and N. M. Burkhead. *Freshwater Fishes of Virginia.* American Fisheries Society, Bethesda, Maryland, 1993.

[90] D. A. Keim. Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, Jan/March 2002.

[91] A. Khotanzan and Y. H. Hong. Invariant Image Recognition by Zernike Moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–487, 1990.

[92] R. Kimmel, D. Shaked, N. Kiryati, and A. M. Bruckstein. Skeletonization via Distance Maps and Level Sets. *Computer Vision and Image Understanding*, 62(3):382–391, 1995.

[93] T. R. Kochtanek and K. K. Hein. Delphi Study of Digital Libraries. *Information Processing and Management*, 35(3):245–254, 1999.

[94] C. Lagoze and H. V. de Sompel. The Open Archives Initiative: Building a Low-Barrier Interoperability Framework. In *Proceedings of the Joint Conference on Digital Libraries*, pages 54–62, Roanoke, VA, USA, June 2001.

[95] A. Lain and J. Fan. Texture Classification by Wavelet Packet Signatures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):186–191, September 1993.

[96] L. J. Latecki and R. Lakamper. Shape Similarity Measure Based on Correspondence of Visual Parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1185–1190, 2000.

[97] M. Leyton. Symmetry-Curvature Duality. *Computer Vision, Graphics, and Image Processing*, 38(3):327–341, 1987.

[98] J. Li, M. Özsu, D. Szafron, and V. Oria. MOQL: A Multimedia Object Query Language. In *Proceedings of the Third International Workshop on Multimedia Information Systems*, pages 19–28, Como, Italy, September 1997.

[99] L.-J. Lin and S.-Y. Kung. Coding and Comparison of DAGs as a Novel Neural Structure with Applications to On-Line Handwriting Recognition. *IEEE Transactions on Signal Processing*, 45(11):2701–2708, 1997.

[100] S. Loncaric. A Survey of Shape Analysis Techniques. *Pattern Recognition*, 31(8):983–1190, Aug 1998.

[101] R. Lotufo and A. Falcão. The Ordered Queue and the Optimality of the Watershed Approaches. In J. Goutsias, L. Vincent, and D. S. Bloomberg, editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 341–350, Palo Alto, USA, Jun 2000. Kluwer Academic.

[102] Y. Lu, H. Zhang, L. Wenyin, and C. Hu. Joint Semantics and Feature Based Image Retrieval Using Relevance Feedback. *IEEE Transactions on Multimedia*, 5(3):339–347, September 2003.

[103] S. D. MacArthur, C. E. Brodley, A. C. Kak, and L. S. Broderick. Interactive Content-Based Image Retrieval Using Relevance Feedback. *Computer Vision and Image Understanding*, 88(2):55–75, 2002.

[104] B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman and Co., San Francisco, 1982.

[105] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan, and A. Yamada. Color and Texture Descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, June 2001.

[106] B. M. Mehtre, M. S. Kankanhalli, and W. F. Lee. Shape Measures for Content Based Image Retrieval: A Comparison. *Information Processing and Management*, 33(3):319–337, 1997.

[107] V. Mesev. Remote Sensing of Urban Systems: Hierarchical Integration with GIS. *Computers, Environment and Urban Systems*, 21(3/4):175–187, 1997.

[108] C. Y. Ming. Shape-Based Image Retrieval in Iconic Image Databases. Master's thesis, Chinese University of Hong Kong, June 1999.

[109] F. Mokhtarian and S. Abbasi. Shape Similarity Retrieval Under Affine Transforms. *Pattern Recognition*, 35(1):31–41, January 2002.

[110] P. Montague and M. Friedlander. Morphogenesis and Territorial Coverage by Isolated Mammalian Retinal Ganglion Cells. *Journal of Neuroscience*, 11:1440–1457, 1991.

[111] E. Moore. The Shortest Path Through a Maze. In *Proc. Intl. Symp. on the Theory of Switching*, pages 285–292. Harvard University, Apr 1959.

[112] H. Muller, W. Muller, D. M. Squire, S. M. Maillet, and T. Pun. Performance Evaluation in Content-Based Image Retrieval: Overview and Proposals. *Pattern Recognition Letters*, 22:593–601, 2001.

[113] A. Nakagawa, A. Kutics, K. Tanaka, and M. Nakajima. Combining Words and Object-based Visual Features in Image Retrieval. In *Proceedings of the 12th International Conference on Image Analysis and Processing*, pages 354–359, Mantova, Italy, September 2003.

[114] S. Nepal, M. Ramakrishna, and J. A. Thom. A Fuzzy Object Query Language (FOQL) for Image Databases. In *Proceedings of the 6th International Conference*

on Database Systems for Advanced Applications, page 117, Hsinchu, Taiwan, April 1999.

[115] V. Noronha. Towards ITS Map Database Interoperability - Database Error and Rectification. *GeoInformatica*, 4(2):201–213, 2000.

[116] OAI. Open Archives Initiative, 2003. http://www. openarchives.org (as of October 2004).

[117] V. E. Ogle and M. Stonebraker. Chabot: Retrieval from Relational Database of Images. IEEE *Computer*, 28(9):40–48, Sep 1995.

[118] OpenGIS. Open GIS Consortium (OGC). http://opengis.org (as of October 2004).

[119] A. Paepcke, C.-C. K. Chang, T. Winograd, and H. Garcia-Molina. Interoperability for Digital Libraries Worldwide. *Communications of the ACM*, 41(4):33–42, 1998.

[120] H. Peitgen, H. Jurgens, and D. Saupe. *Fractals for the Classroom: Introduction to Fractals and Chaos.* Springer Verlag, New York, USA, 1992.

[121] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based Manipulation of Image Databases. In *SPIE Storage and Retrieval for Image and Video Databases II*, pages 34–47, San Jose, CA, 1994.

[122] I. J. PicHunter, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments. *IEEE Transactions on Image Processing*, 9(1):20–37, January 2000.

[123] PlanetMath. http://planetmath.org/ (as of October 2004).

[124] PostGIS. http://postgis.refractions.net (as of October 2004).

[125] PostgreSQL. http://www.us.postgresql.org (as of October 2004).

[126] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, USA, 1985.

[127] P. Rigaux, A. Voisard, and M. Scholl. *Spatial Databases. With Application to GIS.* Morgan Kaufmann, San Fracisco, CA, USA, 2001.

[128] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. Does Organization by Similarity Assist Image Browsing? In *ACM Conference on Human Factors in Computing Systems*, volume 3, pages 190–197, 2001.

[129] Y. Rui, T. S. Huang, and S. F. Chang. Image Retrieval: Current Techniques, Promising Directions, and Open Issues. *Journal of Communications and Image Representation*, 10(1):39–62, March 1999.

[130] J. Sanchez, C. Lopez, and J. Schnase. An Agent-Based Approach to the Construction of Floristic Digital Libraries. In *Proceedings of the 3rd ACM International Conference on Digital Libraries*, pages 210–216, Pittsburgh, PA, July 1998. ACM Press.

[131] J. A. Sanchez, L. Fernandez, and J. L. Schnase. Agora: Enhancing Group Awareness and Collaboration in Floristic Digital Libraries. In *Proceedings of the Fourth International Workshop on Groupware*, pages 85–95, Rio de Janeiro, 1998.

[132] J. A. Sanchez, C. A. Flores, and J. L. Schnase. Mutant: Agents as Guides for Multiple Taxonomies in the Floristic Digital Library. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, pages 244–245, Berkeley, California, USA, 1999. ACM Press.

[133] S. Santini, A. Gupta, and R. Jain. Emergent Semantics through Interaction in Image Databases. *IEEE Transactions on Knowledge and Data Engineering*, 13(3):337–351, May/June 2001.

[134] A. T. Schreiber, B. Dubbeldam, J. Wielemaker, and B. Wielinga. Ontology-Based Photo Annotation. *IEEE Intelligent Systems*, 16:66–74, 2001.

[135] S. Sclaroff, M. L. Cascia, and S. Sethi. Unifying Textual and Visual Cues for Content-Based Image Retrieval on the World Wide Web. *Computer Vision and Image Understanding*, 75(1/2):86–98, July/August 1999.

[136] ShapeDB. www.ee.surrey.ac.uk/research/vssp/imagedb/demo.html, October 2004.

[137] B. Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336–343, Boulder, Colorado, USA, September 1996.

[138] T. Sikora. The MPEG-7 Visual Standard for Content Description - An Overview. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):696–902, June 2001.

[139] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-Based Image Retrieval at the End of the Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, December 2000.

[140] J. R. Smith and S. F. Chang. VisualSEEk: A fully automated content-based image query system. In *Proceedings of the ACM Multimedia*, pages 87–98, Boston, MA, November 1996.

[141] T. R. Smith. A Digital Library for Geographically Referenced Materials. *IEEE Computer*, 29(5):54–60, May 1996.

[142] S.Nepal and M. Ramakrishna. Query Processing Issues in Image (Multimedia) Databases. In *Proceedings of the 15th International Conference on Data Engineering*, pages 22–29, Sydney, Australia, March 23-26 1999.

[143] D. Stan and I. K. Sethi. eID: a System for Exploration of Image Databases. *Information Processing and Management*, 39(3):335–365, 2003.

[144] R. Stehling, M. Nascimento, and A. Falcão. A Compact and Efficient Image Retrieval Approach Based on Border/Interior Pixel Classification. In *Proceedings of the 11th ACM International. Conference on Information and Knowledge Management*, pages 102–109, McLean, Virginia, USA, November 2002.

[145] H. Suleman. *Open Digital Libraries.* PhD thesis, Computer Science Department, Virginia Tech, Blacksburg, VA, 2002. http://scholar.lib.vt.edu/theses/available/etd-11222002-155624/unrestricted/odl.pdf.

[146] H. Suleman, E. Fox, A. Krowne, and M. Luo. Building Digital Libraries from Simple Building Blocks. Technical Report TR-03-09, Computer Science Department, Virginia Tech, Blacksburg, VA, USA, 2003.

[147] H. Suleman, E. A. Fox, R. Kelapure, A. Krowne, and M. Luo. Building Digital Libraries from Simple Building Blocks. *Online Information Review*, 27(5):301–310, 2003.

[148] M. Swain and D. Ballard. Color Indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[149] Y. Y. Tang, Y. Tao, and E. C. M. Lam. New Method for Feature Extraction Based on Fractal Behavior. *Pattern Recognition*, 35(5):1071–1081, 2002.

[150] Q. Tian, B. Moghaddam, and T. S. Huang. Display Optimization for Image Browsing. In *Proceedings of the 2nd International Workshop on Multimedia Databases and Image Communications*, Amalfi, Italy, Sep 2001.

[151] C. Traina, B. Seeger, C. Faloutsos, and A. Traina. Fast Indexing and Visualization of Metric Datasets Using Slim-Trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(2):244–60, March/April 2002.

[152] C. Tricot. *Curves and Fractal Dimension*. Springer-Verlag, New York, USA, 1995.

[153] K. Vu, K. A. Hua, and W. Tavanapong. Image Retrieval Based on Regions of Interest. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):1045–1049, July/August 2003.

[154] T. P. Wallace and P. Wintz. An Efficient Three-dimensional Aircraft Recognition Algorithm Using Normalised Fourier Descriptors. *Computer Graphics Image Processing*, 13:99–126, 1980.

[155] S. Wang and H. Wang. Knowledge Discovery Through Self-Organizing Maps: Data Visualization and Query Processing. *Knowledge and Information Systems*, 4(1):31–45, 2002.

[156] Y. P. Wang and T. Pavlids. Optimal Correspondence of String Subsequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(11):1080–1087, 1990.

[157] M. Weber, M. Alexa, and W. Muller. Visualizing Time-Series on Spirals. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 7–14, San Diego, California, USA, 2001.

[158] WFS. Web Feature Server (WFS). http://www.opengis.org/specs (as of October 2004).

[159] XMLFile. http://www.dlib.vt.edu/projects/OAI/software/oai-file/oai-file.html (as of October 2004).

[160] XMLSpy. http://www.xmlspy.com/ (as of October 2004).

[161] K.-P. Yee, D. Fisher, R. Dhamija, and M. A. Hearst. Animated Exploration of Dynamic Graphs with Radial Layout. In *Proceedingsof the IEEE Symposium on Information Visualization*, pages 43–50, San Diego, California, USA, 2001.

[162] D. Zhang and G. Lu. Review of Shape Representation and Description. *Pattern Recognition*, 37(1):1–19, Jan 2004.

[163] R. Zhao and W. I. Grosky. Narrowing the Semantic Gap – Improved Text-Based Web Document Retrieval Using Visual Features. *IEEE Transactions on Multimedia*, 4(3):189–200, June 2002.

[164] X. S. Zhou and T. S. Huang. Unifying Keywords and Visual Contents in Image Retrieval. *IEEE Multimedia*, 4(2):23–33, June 2002.

[165] B. Zhu, M. Ramsey, and H. Chen. Creating a Large-Scale Content-Based Airphoto Image Digital Library. *IEEE Transactions on Image Processing*, 9(1):163–167, January 2000.