Descrição de Texturas Invariante à Rotação e Escala para Identificação e Reconhecimento de Imagens

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Javier Alexander Montoya Zegarra e aprovada pela Banca Examinadora.

Campinas, 21 de Dezembro de 2007.

Neucimar J. Leite (Orientador)
Instituto de Computação
Universidade Estadual de Campinas

Ricardo Torres
Ricardo da Silva Torres (Co-orientador)
Instituto de Computação
Universidade Estadual de Campinas

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

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP
Bibliotecária: Miriam Cristina Alves – CRB8a / 5094

Montoya Zegarra, Javier Alexander
M768d Descrição de texturas invariante à rotação e escala para identificação e reconhecimento de imagens/Javier Alexander Montoya Zegarra -- Campinas, [S.P. :s.n.], 2007.

Orientadores : Neucimar Jerônimo Leite; Ricardo da Silva Torres
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Análise de imagem. 2. Processamento de imagens. 3. Banco de dados. 4. Sistemas de recuperação de informação. I. Leite, Neucimar Jerônimo. II. Torres, Ricardo da Silva. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Título em inglês: Rotation-invariant and scale-invariant texture representation for image identification and recognition.

Palavras-chave em inglês (Keywords): 1. Image analysis. 2. Image processing. 3. Databases. 4. Information retrieval systems.

Área de concentração: Ciência da Computação

Titulação: Mestre em Ciência da Computação

Banca examinadora: Prof. Dr. Neucimar Jerônimo Leite (IC-Unicamp)
Profa. Dra. Agma Juci Machado Traina (ICMC-USP)
Prof. Dr. Siome Klein Goldenstein (IC-Unicamp)

Data da defesa: 21/12/2007

Programa de Pós-Graduação: Mestrado em Ciência da Computação

TERMO DE APROVAÇÃO

Dissertação Defendida e Aprovada em 21 de dezembro de 2007, pela Banca examinadora composta pelos Professores Doutores:

Profª. Drª. Agma Juci Machado Traina
ICMC - USP

Prof. Dr. Siome Klein Goldenstein
IC - UNICAMP

Prof. Dr. Neucimar Jerônimo Leite
IC - UNICAMP

Instituto de Computação
Universidade Estadual de Campinas

Descrição de Texturas Invariante à Rotação e Escala para Identificação e Reconhecimento de Imagens

Javier Alexander Montoya Zegarra[1]

Dezembro de 2007

Banca Examinadora:

- Neucimar J. Leite (Orientador)
  Instituto de Computação
  Universidade Estadual de Campinas
- Ricardo da Silva Torres (Co-orientador)
  Instituto de Computação
  Universidade Estadual de Campinas
- Agma Juci Machado Traina
  Instituto de Ciências Matemáticas e de Computação
  Universidade de São Paulo
- Siome Klein Goldenstein
  Instituto de Computação
  Universidade Estadual de Campinas
- Alexandre Xavier Falcão (Suplente)
  Instituto de Computação
  Universidade Estadual de Campinas

[1]**Suporte financeiro do CNPq, número do processo 134990/2005-6.**

Resumo

Uma importante característica de baixo nível, utilizada tanto na percepção humana como no reconhecimento de padrões, é a textura. De fato, o estudo de textura tem encontrado diversas aplicações abrangendo desde segmentação de textura até síntese, classificação e recuperação de imagens por conteúdo.

Apesar das múltiplas técnicas eficientes e eficazes propostas para classificação e recuperação, ainda há alguns desafios que precisam ser superados como, por exemplo, a necessidade de descritores de imagens compactos e robustos a serem empregados na consulta e classificação de bases de imagens de textura.

Esta dissertação propõe um descritor de imagens de textura visando à busca e à recuperação de bases de dados de imagens. Este descritor baseia-se na Decomposição Piramidal *Steerable* caracterizada por sua análise de forma invariante à rotação ou à escala. Resultados preliminares conduzidos em cenários não-controlados demonstraram caráter promissor da abordagem.

No que diz respeito à classificação de imagens de textura, esta dissertação propõe ao mesmo tempo um sistema de reconhecimento, o qual possui como principais características representações compactas de imagens e módulos de reconhecimento eficientes. O descritor proposto é utilizado para codificar a informação relevante de textura em vetores de características pequenos. Para tratar os requisitos de eficiência do reconhecimento, uma abordagem multi-classe baseada no classificador de Floresta de Caminhos Ótimos é utilizada. Experimentos foram conduzidos visando avaliar o sistema proposto frente a outros métodos de classificação. Resultados experimentais demonstram a superioridade do sistema proposto.

Abstract

An important low-level image feature used in human perception as well as in recognition is texture. In fact, the study of texture has found several applications ranging from texture segmentation to texture synthesis, classification, and image retrieval.

Although many efficient and effective techniques have been proposed for texture classification and retrieval, there are still some challenges to overcome. More specifically, there is a need for a compact and robust image descriptor to query and classify texture image databases.

In order to search and query image databases, this dissertation provides a texture image descriptor, which is based on a modification of the Steerable Pyramid Decomposition, and is also characterized by its capabilities for representing texture images in either rotation-invariant or scale-invariant manners. Preliminary results conducted in non-controlled scenarios have demonstrated the promising properties of the approach.

In order to classify texture images, this dissertation also provides a new recognition system, which presents as main features, compact image representations and efficient recognition tasks. The proposed image descriptor is used to encode the relevant texture information in small size feature vectors. To address the efficiency recognition requirements, a novel multi-class object recognition method based on the Optimum Path Forest classifier is used. To evaluate our proposed system against different methods, several experiments were conducted. The results demonstrate the superiority of the proposed system.

Agradecimentos

O que seria de nossas vidas sem um sentido para viver? Se descobríssemos que todos os nossos questionamentos mais profundos não têm respostas e tudo é apenas casualidade? Assim, eu gostaria de iniciar essa seção de Agradecimentos começando pela origem da vida, da minha vida, da tua vida, da nossa vida. Quero te agradecer “Taita” pelo mistério da vida e da existência, por permitir ir te conhecendo e te encontrando aos poucos nas outras pessoas. Por permitir, ao mesmo tempo, me conhecer e me entender melhor através de ti. Obrigado pelas bênçãoes recebidas, pessoas conhecidas e histórias confiadas.

Eu quero agradecer aos meus pais, Javier Montoya Gómez e Julie Zegarra Farach, pelo seu exemplo de doação e sacrifício. Por cuidar com dedicação dos seus filhos e vibrar com cada um dos nossos passos. À minha irmã Ashraf (Asharé) pela sua paciência, compreensão e carinho. A cada um dos membros da minha familia: avós, tios, primos e sobrinhos, por permitir que a distância e às vezes os poucos momentos compartilhados passem despercibidos tanto no espaço como no tempo e não sejam limites para o nosso carinho. Cada um deles, desde o mais novos até os mais adultos, me deixaram ensinamentos muito presentes na minha vida. Obrigado pelas histórias e momentos compartilhados! Eu gostaria de agradecer em especial ao meu tio Jorge Montoya, pelo seu exemplo de entrega e por acompanhar minuciosamente cada uma das nossas atividades ao longo desse tempo todo.

An die Familie Kozel aus Straubenhardt fuer die ganze Unterstuetzung in diesen ganzen Jahren, sowie fuer die wunderbare und unvergessbare Momente in Deutschland. Danke auch an die Lehrer und Kameraden der Gymnasium Neuenbuerg und VFR Pforzheim.

Aos meus sempre amigos da *generation perdue* por permitir que os momentos vividos juntamente não fiquem apenas em lembranças, mas também, que continuem sendo vividos ainda no presente apesar da distância. Obrigado a Gonzalo Valdivia, Jan Beeck, Johannes Textor, Mauricio Azálgara e Victor Carbajal!

Aos meus amigos e colegas do colégio peruano-alemão Max-Uhle. Em especial aos membros da turma *Vereinte Kraft 1998* pelas tantas anedotas vividas.

Ao meu *Oh Captain, my Captain* Javier Rodriguez, bem como a Oscar Osterling, Gilmer e Fernando Genú pelo seu empenho e acompanhamento no meu crescimento humano. Aos meus irmãos de *agrupa* pelas tantas frutíferas reuniões e *doações*. Aos membros da família do Movimento de Vida Cristã pelos momentos compartilhados.

A época de graduação na Universidade Católica San Pablo foi muito grata e especial. Devo isso tudo a várias pessoas, começando pelas pessoas do setor administrativo até os amigos, colegas e professores. Entre outras pessoas gostaria de agradecer com bastante carinho a Carla Cornejo, Carmen Carpio, Claudia Jara, Jorge Perochena, José Molina, Kelvin Macedo, Keyvin Campos, Marco Antonio Vásquez, Talia Leiva e Valery Banda. Sem a ajuda, partilhas e experiências com eles, a época da graduação não seria o que ela representa para mim. Muitas dúvidas foram surgindo durante a vida universitária e sem a ajuda e conversas com Alonso Quintanilla e Manuel Rodriguez estas não teriam sido esclarecidas. Aos professores e grandes amigos Juan Carlos Gutierrez e Alex Cuadros por acreditar no meu trabalho, pelo exemplo de simplicidade, desejos de superação e de compromisso com a realidade. Aos professores Luis Diaz, Wilber Ramos, Eduardo Tejada e Ernesto Cuadros pelos conhecimentos compartilhados. Essa comunidade toda representa para mim um bom lugar de desenvolvimento pessoal e profissional acompanhado de anédotas e histórias que merecem ser relembradas.

Aos membros do time *viventura Reisen GmbH*, pois foi lá que tive a minha primeira experiência professional. O dia-a-dia de convívio com um time jovem, dinâmico e multicultural me ajudaram a amadurecer profissionalmente. Os momentos fora do trabalho como viagens, projetos sociais e reuniões, foram instâncias para me desenvolver como pessoa. Assim gostaria de agradecer a todos os funcionários por esses quase 4 anos de experiências inesquecíveis.

Nesses quase 3 anos de Brasil tenho que agradecer a todas as pessoas que através do seu apoio fizeram que esse tempo fique com boas lembranças. Eu gostaria de começar agradecendo a Felipe Klein, Patrick Brito e Evandro Bacarin pelos exemplos de dedicação e de humanidade. Aos dois primeiros também pelo clima de irmandade no tempo que morei com eles, bem como pelas alegrias, piadas e companhia nos momentos mais fracos (deu para entender :-)?). Obrigado por essa amizade toda! Ao terceiro por ter se preocupado pelo meu constante crescimento pessoal (mesmo apesar da distância). Gostaria também de agradecer a André Santanchè e Luis Celso pela acolhida desde a minha che-

gada ao Brasil. Por compartilhar comigo elementos da “cultura brasileña” e sobretudo pelos conselhos e bate-papos.

O que teria sido dessa experiência no Brasil sem uma *chiquitita*? Obrigado Vanessa por cada um dos momentos vividos. Pelo seu carinho, ensinamentos, confiança e por me ajudar a ser uma melhor pessoa.

Aos meus orientadores Neucimar Leite e Ricardo Torres por terem acreditado no meu trabalho e por terem me dado a liberdade de conduzir a presente pesquisa. Também pelas frutíferas reuniões pessoais e profissionais, bem como pela abertura ao diálogo e comprometimento no desenvolvimento deste trabalho (mesmo em horários inoportunos ;-)).

No contexto do quase dia-a-dia, ou às vezes, noite-a-noite de atividades profissionais, gostaria de agradecer aos colegas do Laboratório de Sistemas de Informação (LIS) pelo apoio e pelos momentos compartilhados fora e dentro dele. Em especial à profa. Claudia Bauzer Medeiros bem como aos companheiros: Alan Nakai, Andréia Kondo, Carla Macário, Cristiano Dalmaschio, Fernando Ito, Gilberto Zonta, Jaudete Daltio, Jefersson dos Santos, Joana Malaverri, Leonardo Mariote, Luciano Digiampietri, Márcio Marar, Ricardo Freitas, Rodrigo Martins, Rodrigo Senra, Otávio Penatti e Wesley Mesquita. Aos membros do grupo de pesquisa do professor Neucimar, pelas idéias e dicas de pesquisa bem como pelos momentos alheios à área acadêmica: Claudio Raimundo, Fábio Dias, Leandro Marco, Leyza Dorini, Luis Delval, Nielsen Simões, Pascual Figueroa, Rafael Castanheira e Rodrigo Minetto. A Neumar Malheiros e Tiago Senador por aqueles gratos momentos vividos, bem como pelos bate papos. Aos colegas e companheiros do Instituto de Computação (IC), aos professores, às pessoas do setor administrativo e de serviço por fazer dele um bom lugar onde se estudar e se desenvolver.

Aos professores e pesquisadores que colaboraram com o estudo realizado nesta dissertação: Alexandre Falcão, Craig Watson, Gustavo de Sá, Heikki Ailisto, João Papa, Kenneth Nilsson, Marcelo Oliveira, Nilceu Marana, Pascual Figueroa, Serge Belongie e Sharat Chikkerur.

Às pessoas que me apoiaram durante o meu mestrado. Gostaria de agradecer em especial a: Markus Mock, Michael Grubinger e ao padre José Luís Araújo pelos seus conselhos dados bem como pelos seus desejos em me ajudar.

Aos “brasileños” com os quais morei e que me permitiram conhecer mais de perto a cultura brasileira. Gostaria de agradecer em especial aos membros das repúblicas 999 e Cachorros Magros: André Asti, Ednei Amaral, Elizabeth Baptestini, Fábio Lumertz, Francisco Augusto, Gabriel Castiglioni, Luiz Lacerda, Luiz Ywata, Matheus Tait, Maurício Andrade, Rodrigo Galbieri, Tiago Nunes e Valdir de Assis.

Aos membos do Centro Cultural do Castelo pelas tertúlias e experiências tidas. Em especial aos padres Cesário e Luiz, a Elber, Francesco, Gentil, Miguel, Sérgio e Waltão.

Aos membros do grupo de jovens da paróquia Santa Isabel, pelos ensinamentos, pela confiança e por me deixarem ir conhecendo a cada um deles melhor. Por representar um espaço de crescimento pessoal.

Ao CNPq, Fapesp, Faepex, Capes e Microsoft pelo apoio e suporte financeiro durante o desenvolvimento do presente trabalho.

Concluo finalmente esta seção de Agradecimentos com um grato obrigado a todas as pessoas anônimas que passaram pela minha vida e que me deixaram algo de si.

"Duc in altum!" - "Put out into the deep!" (*Lc 5,4*)

To *Don Lucito*.

Sumário

Lista de Tabelas

Lista de Figuras

Capítulo 1

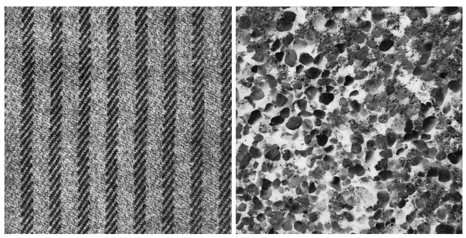
Introdução

Uma importante característica de baixo nível, utilizada tanto na percepção humana como no reconhecimento de padrões, é a textura. Embora a percepção de textura seja intuitiva (pois é possível reconhecê-la ao enxergá-la), ainda não existe uma definição precisa sobre a mesma. Neste sentido, a literatura tem apresentado diferentes definições. Dentre as quais podemos citar:

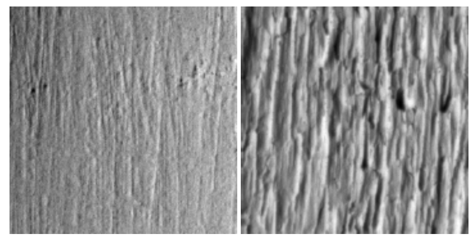
- "Uma imagem de textura é descrita pelo número e tipos de suas primitivas (tonais), e a organização espacial ou distribuição de suas primitivas (tonais)..." [47].
- "A textura é considerada como aquilo que constitui uma região macroscópica. Sua estrutura é simplesmente atribuída aos padrões repetitivos nos quais elementos ou primitivas estão organizados de acordo com uma regra de colocação..." [129].
- "A noção de textura parece depender de três ingredientes: (*i*) alguma 'ordem' local é repetida sobre uma região que é grande em relação ao tamanho da ordem, (*ii*) a ordem consiste em um arranjo não aleatório de partes elementares, e (*iii*) as partes são geralmente entidades uniformes que possuem aproximadamente as mesmas dimensões em toda parte dentro da região de textura." [48].
- "A propriedade básica de uma textura é um padrão pequeno elementar que é repetido periodicamente ou quase-periodicamente em um espaço como um padrão em um papel de fundo. Assim, é suficiente descrever o padrão elementar pequeno e as regras de repetição..." [54].

Embora exista uma grande diversidade de definições, ainda não há um conceito universalmente aceito sobre o que seja uma textura. Contudo, estas definições apresentam uma série de pontos em comum [123]: (1) numa mesma textura existe uma grande variação nos níveis de intensidade entre pixels próximos, isto é, no limite da resolução, há não-homogeneidade e (2) a textura é uma propriedade homogênea em alguma escala espacial maior que a resolução da imagem.

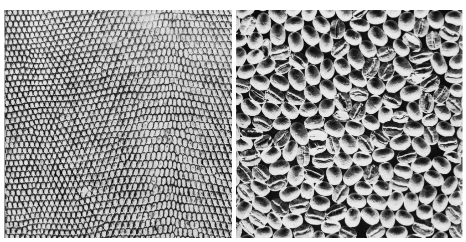
A homogeneidade nas texturas, representada pela repetição de padrões gera uma série de características visuais, que podem ser identificadas como sendo, por exemplo, direcionais, finas, ásperas e uniformes [71,129,145]. Alguns exemplos deste tipo de características visuais podem ser encontrados nas Figuras 1.1(a)- 1.1(d) [146]. Note que cada textura pode estar associada a uma ou mais destas características.



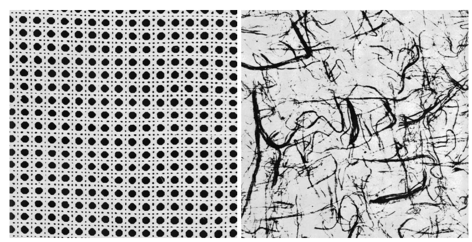
(a) Características visuais *direcionais vs. não-direcionais.*



(b) Características visuais *suaves vs. não-suaves.*



(c) Características visuais *finas vs. ásperas.*

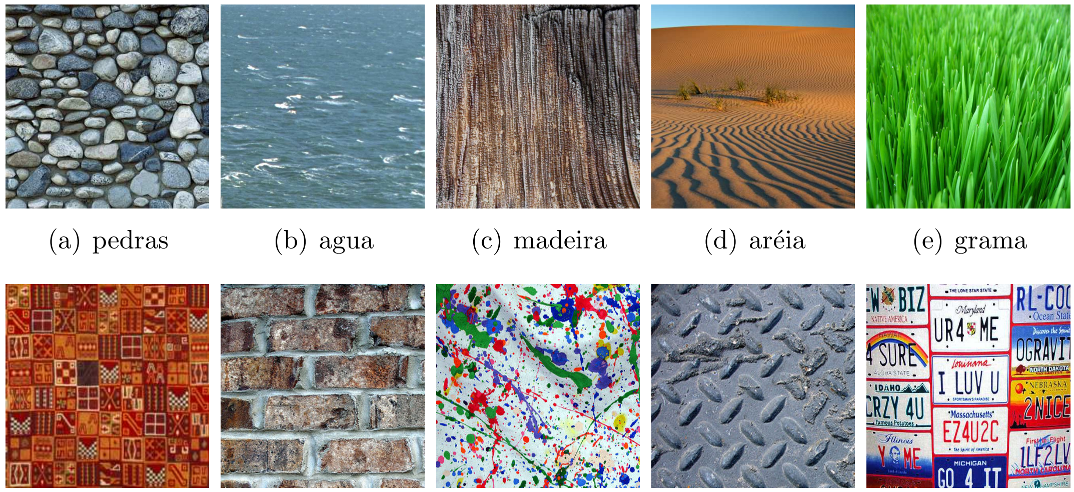


(d) Características visuais *uniformes vs. não-uniformes.*

Figura 1.1: Alguns tipos de características visuais de textura.

Por outro lado, as imagens de textura são tipicamente classificadas como sendo **naturais** ou **artificiais**. As texturas naturais estão relacionadas a objetos que não são produzidos pelo homem. Exemplos desta classe de texturas incluem: pedras, água, madeira, areia e grama. Por outro lado, nas texturas artificias existe uma intervenção do homem. Alguns exemplos incluem padrões têxteis, construção, pintura, metal. Algumas amostras de texturas naturais e artificiais são apresentadas na Figura 1.2.

Independentemente do seu tipo de classificação, as imagens de textura podem ser caracterizadas pelas suas variações em escala e direcionalidade. As variações na escala implicam que texturas do mesmo tipo podem parecer diferentes uma das outras quando apresentadas em diferentes escalas. Este efeito é análogo ao incrementar ou decrementar a resolução da imagem.



(a) pedras (b) agua (c) madeira (d) aréia (e) grama

(f) tecido inca (g) parede de tijolos (h) pintura (i) metal (j) placas de carro

Figura 1.2: Exemplos de texturas naturais e artificiais. A primeira linha apresenta 5 diferentes tipos de texturas naturais. A segunda linha apresenta 5 diferentes tipos de texturas artificiais.

Esta propriedade está relacionada com a granuralidade (*coarseness*) presente nas imagens de textura pode ser entendida como o período de repetição espacial do padrão local [116]. As imagens de textura mais finas apresentam períodos de repetição menores, enquanto que as texturas de maior granuralidade apresentam períodos mais acentuados de repetição. Além disso, texturas da mesma classe, caracterizadas por orientações definidas, podem apresentar diferentes direções dominantes quando a mesma imagem sofre rotações. Isto acontece entre outros, devido aos diferentes pontos de vista e de captura das imagens.

1.1 Análise de textura

A análise de textura é uma área que vem despertando grande interesse há alguns anos por diversos grupos de pesquisa e de indústria espalhados pelo mundo. Como resultado destes estudos, diversas aplicações foram propostas, incluindo desde tarefas de recuperação de imagens por conteúdo [76, 125] até tarefas de classificação [136], segmentação [120], síntese [6, 147] e obtenção de forma a partir de textura [78] (veja Figura 1.3). Essas cinco principais aplicações podem ser catalogadas como as principais áreas de pesquisa no campo de análise de textura [86]. Cada uma delas será discutida brevemente a seguir.

Estas aplicações caracterizam-se por um elemento em comum. Todas procuram responder a pergunta de como caracterizar/representar uma textura de forma que as suas propriedades mais importantes sejam extraídas. O foco desta dissertação se encontra na obtenção de métodos de caracterização de texturas para fins de recuperação de imagens por conteúdo e classificação. Além disso, uma importante motivação do nosso estudo é a de utilizar vetores de características de baixa dimensão, visando facilitar aplicações nas quais o espaço de armazenamento dos mesmos representa uma limitação.

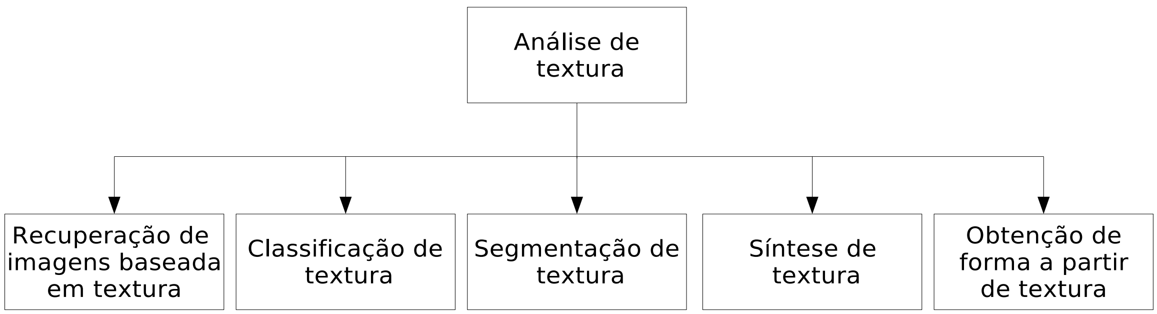


Figura 1.3: Áreas principais de análise de textura.

1.1.1 Recuperação de imagens por conteúdo

Devido ao grande número de imagens coletadas a cada dia nas áreas civis, comercias e acadêmicas, bem como aos avanços nas tecnologias de aquisição e digitalização de imagens, os bancos de dados de imagens caracterizam-se hoje em dia pelo seu grande volume. Neste sentido, a idéia das técnicas de recuperação de imagens por conteúdo baseadas em textura consiste em oferecer mecanismos que possibilitem a indexação dessas bases. Para tanto, utilizam-se características discriminativas texturais que representam o conteúdo visual das imagens capturadas por meio de descritores. Tradicionalmente, um descritor é composto por: (i) um algoritmo de extração de características que gera um vetor de características, e por (ii) uma função de distância utilizada para se calcular a distância entre duas imagens. Neste sentido, para se obter a similaridade entre um par de imagens, calcula-se a distância entre os seus vetores de características. Assim, as imagens podem ser pesquisadas utilizando-se representações visuais, sendo que as imagens recuperadas são as visualmente mais similares à imagem de busca. Alguns exemplos deste tipo de sistemas na literatura incluem: Photobook [110], VisualSEEk [126] e Cortina [39, 119].

No contexto de recuperação de imagens por conteúdo, a textura pode ser utilizada de duas maneiras. No primeiro caso, as características texturais representativas das imagens são extraídas e depois armazenadas em uma base de dados, de tal forma que possam ser utilizadas posteriormente na busca de imagens mais semelhantes considerando-se uma imagem de consulta [66, 80, 92]. As imagens utilizadas podem ser: (*i*) regiões de texturas obtidas após um processo de segmentação, (*ii*) imagens de texturas particionadas em subimagens, ou (*iii*) imagens de textura sem nenhum pré-processamento.

No segundo caso, a textura é utilizada para anotação de imagens [101,112,140]. A idéia consiste em fornecer aos usuários anotações automáticas das imagens com base nas suas propriedades de textura. Essa tarefa é muito importante, sobretudo quando o conjunto de imagens a ser anotado é muito grande. Dada uma anotação inicial fornecida pelo usuário, o sistema é capaz de gerar um modelo que pode ser aplicado a outros tipos de imagens similares. Dessa forma, a subjetividade na anotação das imagens pode ser reduzida.

Uma importante característica dos sistemas de recuperação de imagens por conteúdo é que ao invés de se obter uma única imagem, um conjunto de N imagens pode ser encontrado. Neste caso, cada uma das imagens é ordenada em relação à sua similaridade com a imagem de consulta. Note que a definição da quantidade de imagens a ser recuperada, representada pela variável N, é flexível com relação aos requisitos do sistema.

1.1.2 Classificação de textura

O objetivo da classificação de textura consiste em atribuir uma dada imagem *desconhecida* a uma dada categoria de um conjunto de classes de texturas *conhecidas*. Tradicionalmente, a tarefa de classificação envolve as fases de aprendizado e reconhecimento [58, 121].

Na fase de aprendizado, procura-se criar um modelo de cada classe de textura presente nos dados de treinamento. Assim, utilizando-se um método de representação de textura que busca caracterizar suas propriedades mais relevantes, extrai-se um conjunto de vetores de características para cada uma das imagens. Normalmente estes vetores de características consistem em números escalares que representam algumas propriedades das texturas tais como orientação, luminosidade, contraste, etc.

Já na fase de reconhecimento, utiliza-se o mesmo método de representação textural para se extrair o vetor de características da imagem a ser classificada. Em seguida, um algoritmo de classificação é utilizado para se comparar o vetor de características extraído com aqueles obtidos das imagens de treinamento. Assim, considerando-se o melhor casamento, a classe da textura pode ser atribuída. Os possíveis resultados da classificação podem ser a correta classificação da imagem ou a rejeição da classe indicada. Além disso, a pesquisa recente neste tópico direciona-se também para o reconhecimento de texturas invariantes a deformações geométricas, tais como rotação e escala [74,96,98,118].

Nesta dissertação, apenas a classificação de texturas supervisionada é considerada e a acurácia da classificação refere-se à porcentagem das amostras corretamente classificadas.

1.1.3 Segmentação de textura

O objetivo da segmentação de textura é decompor uma dada imagem em um conjunto disjunto de componentes nos quais a textura é constante [36, 55, 68, 84]. A maioria dos métodos existentes na literatura limita-se ao problema de segmentação de texturas em níveis de cinza [12, 133, 135]. Avanços nessa última década referem-se à abordagem do problema de segmentação de texturas de imagens coloridas [91,127,141]. Alguns exemplos de imagens segmentadas por textura são apresentados na Figura 1.4.

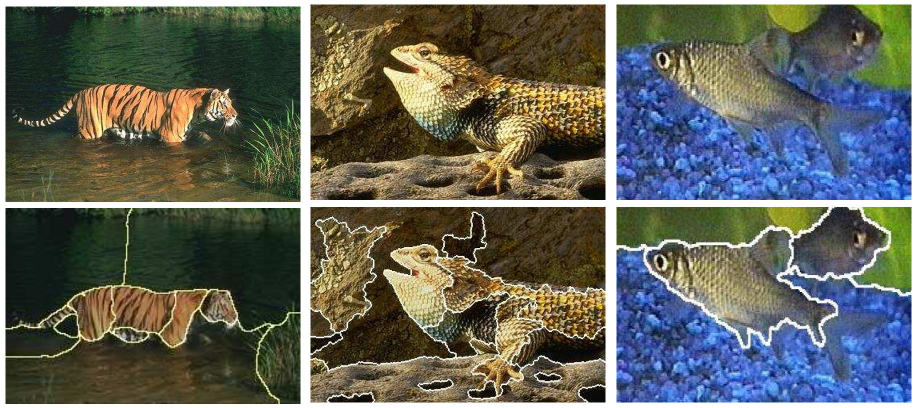
Tradicionalmente, a tarefa de segmentação é composta por duas atividades: (*i*) representação da textura e (*ii*) determinação das fronteiras de segmentação. A idéia baseia-se no fato de que para se determinar se duas dadas texturas são iguais, precisa-se de um método de representação da textura de forma que as fronteiras adjacentes de regiões texturais diferentes possam ser identificadas.

A existência de uma informação *a priori* dos tipos de textura presentes numa imagem implica uma segmentação dita *supervisionada*. Caso os pixels ou regiões tenham que ser agrupados com base numa medida de similaridade perceptual, a segmentação será dita *não supervisionada*. Além disso, os métodos de segmentação podem ser classificados como sendo baseados em regiões ou contornos (*boundary*). A idéia dos métodos de segmentação com base em regiões é agrupar pixels ou regiões locais menores relativos à similaridade de alguma propriedade textural. Dado que neste tipo de método pode ser necessário especificar o número de texturas diferentes na imagem, os mesmos podem ser considerados como métodos de segmentação *supervisionados*. Adicionalmente, os métodos baseados na detecção de contornos subdividem a imagem em fronteiras nas quais existe alguma diferença textural entre regiões adjacentes. Este tipo de segmentação é *não supervisionada*, já que não necessita do conhecimento a priori do número de diferentes texturas da imagem.

1.1.4 Síntese de Textura

O objetivo principal da síntese de textura é gerar uma imagem de tamanho arbitrário a partir de uma amostra de textura [73, 143]. Em outras palavras, o problema pode ser entendido como: dada uma amostra de textura, sintetize-a em uma nova textura de tal forma que, quando percebida por um observador humano, esta aparente ter sido gerada pelo mesmo processo estocástico básico [142]. Os dois grandes desafios nesta tarefa consistem, por um lado, em como modelar o processo estocástico de uma dada textura finita e, por outro, em como desenvolver de maneira eficiente um procedimento de amostragem para gerar a nova imagem. A fidelidade da representação sintética obtida depende da modelagem utilizada, enquanto que o custo computacional depende da eficiência da amostragem.

Os métodos de síntese de textura mais comuns que visam sintetizar as imagens em



(a) (b) (c)

Figura 1.4: Exemplos de alguns métodos de segmentação de textura: (a) Martin et al., 2004 [90], (b) Deng e Manjunath, 2001 [29], (c) Allili e Ziou, 2007 [2].

malhas 2D podem ser divididos em dois grandes grupos [81]: os baseados em pontos e os baseados em blocos (*patches*). Os métodos baseados em pontos procuram sintetizar as texturas colorindo um pixel a cada vez. Para se determinar a cor no pixel, procura-se na amostra de textura por aquele pixel cujos vizinhos mais próximos possuem cores semelhantes aos que se encontram na vizinhança atual [134, 142]. Os métodos baseados em blocos procuram sintetizar trechos maiores de textura copiando regiões selecionadas da amostra e ocultando de alguma maneira as fronteiras. Geralmente o proceso de seleção de blocos é aleatório de tal modo que uma imagem de alta qualidade pode ser obtida sem a necessidade de padrões visuais não reais [102, 117].

1.1.5 Forma a partir de textura

Experimentos psisicológicos têm demostrado que a textura é muito importante na percepção humana de formas (*shapes*) [14, 26, 128]. Neste sentido, o objetivo principal das técnicas de obtenção de forma a partir de textura (*shape from texture*) consiste em obter a orientação da superfície ou a forma da mesma (forma $3D$) a partir de um objeto textural contido numa imagem [10, 23, 36, 132].

Os métodos que recuperam modelos de superfícies a partir da projeção de uma textura podem ser classificados em dois grandes grupos [35]: globais e locais. Métodos globais procuram recuperar um modelo da superfície inteiro utilizando-se para tais fins a distribuição dos elementos de textura. Por outro lado, os métodos locais recuperam alguns parâmetros característicos da geometria de um dado ponto na superfície. Os parâmetros característicos incluem entre outros as curvaturas e a normal dos pontos.

Independentemente do seu tipo de classificação, uma parte básica dos métodos de forma a partir de textura são os elementos de textura. Um elemento de textura é aquele caracterizado pela sua (i) repetição, isto é, um elemento tem que ser repetido o número suficiente de vezes para prover uma informação útil da geometria da superfície, e (ii) localidade, significando que a estrutura do elemento pode ser encontrada na imagem e utilizada para inferir uma transformação de profundidade [78].

Figura 1.5: Exemplos de recuperação de forma de superfícies em padrões têxteis. Resultados obtidos em Lobay e Forsyth, 2006 [78].

1.2 Objetivos

A forma de como a análise de textura é abordada nesta dissertação consiste no estudo de métodos que possibilitem representações compactas e eficazes para capturar as propriedades relevantes em imagens de texturas. Além disso, nenhuma restrição é feita quanto à natureza das imagens. Dessa forma, pode-se garantir que o problema de caracterização da textura seja mantido o mais genérico possível.

Os ambientes *não-controlados* considerados nesta dissertação estão geralmente caracterizados por imagens de texturas com variações inter-classes e distorções geométricas. As variações inter-classes estão relacionadas com o fato de que as texturas de diferentes classes podem ser bem semelhantes. Este aspecto é ilustrado na Figura 1.6.

Por outro lado, as distorções presentes nas imagens de textura referem-se tanto a orientações bem como escalas, podendo gerar novas caracterizações de texturas diferentes das imagens originais. A Figura 1.7 apresenta alguns exemplos destes tipos de distorções.

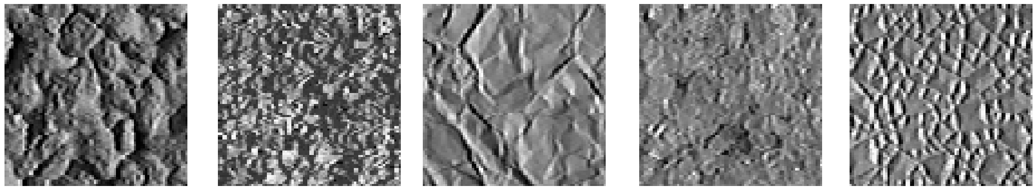


Figura 1.6: Exemplos de inter-classes entre imagens de textura. Apesar da sua similaridade visual, cada uma das imagens apresentadas correspondem a classes diferentes.

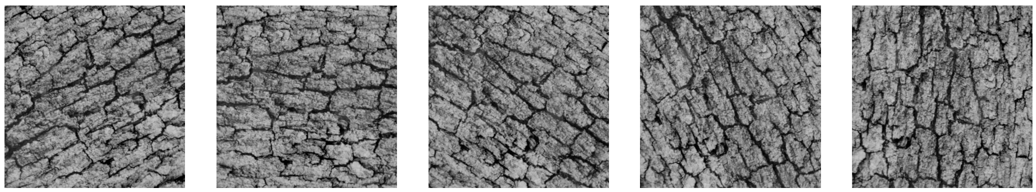


Figura 1.7: Exemplos de imagens de textura nas quais existem rotações.

1.3 Contribuições

Além do estudo teórico de tópicos sobre análise de textura, bem como o amplo levantamento bibliográfico de métodos, o presente trabalho considerou o estudo preliminar e implementação de abordagens para identificação e reconhecimento de imagens com base em sua textura. Neste contexto, ele possui as seguintes contribuições:

- Proposta de um novo descritor de textura invariante à rotação e à escala visando aplicações de recuperação de imagens por conteúdo. Este descritor é baseado na Decomposição Piramidal *Steerable* (*Steerable Pyramid Decomposition*). Para avaliar a acurácia do descritor proposto, comparações com um método baseado em *Gabor Wavelets* foram realizadas [87]. A partir dos resultados obtidos pôde-se demostrar a superioridade do método proposto. Maiores detalhes sobre o descritor podem ser encontrados no Capítulo 2 e no artigo [92].

- Proposta e avaliação de um novo sistema eficaz de reconhecimento de texturas invariante à rotação e à escala. Trata-se de um método que considera ambos os processos de extração de características e de classificação. Para tais fins, utiliza-se o descritor proposto bem como o recente classificador baseado no Caminho Ótimo de Floresta (OPF - Optimum Path Forest) [107]. Maiores detalhes encontram-se nos Capítulos 3 e 4, bem como nos artigos [96, 97].

- Proposta e avaliação de um novo esquema eficiente para identificação de impressões digitais. Trata-se de um método que considera ambos extração de características e cálculo de similaridade, utilizando-se para tais fins diferentes tipos de Wavelets e métricas de similaridade. Maiores detalhes encontram-se no Capítulo 5 e nos artigos [93–95, 148].

1.4 Organização da dissertação

A presente dissertação está estruturada considerando-se os principais artigos publicados e/ou submetidos para publicação no decorrer dos estudos realizados. A ordem de apresentação dos capítulos foi estruturada de maneira que exista uma conexão lógica entre os mesmos. Neste sentido, a seqüência não representa necessariamente a ordem de publicação dos artigos.

O Capítulo 2 apresenta contribuições no domínio de recuperação de imagens por conteúdo, enquanto que os capítulos 3 e 4 tratam de contribuições em visão computacional, mais especificamente na área de reconhecimento de padrões. Finalmente, o capítulo 5 apresenta resultados em biometria e o capítulo 6 conclui a dissertação.

1.4.1 *Rotation-Invariant and Scale-Invariant Steerable Pyramid Decomposition for Texture Image Retrieval*

O Capítulo 2 (*Rotation-Invariant and Scale-Invariant Steerable Pyramid Decomposition for Texture Image Retrieval*) inclui os resultados apresentados em [92]. Nesse trabalho apresentam-se os resultados referentes à proposta de um novo descritor de textura utilizado no suporte à recuperação de imagens por conteúdo.

O cálculo deste descritor utiliza a Decomposição Piramidal *Steerable*, a qual consiste num método de decomposição multi-resolução em que uma imagem é decomposta num conjunto de sub-bandas multi-orientação e multi-escala onde as funções de base são operadores direcionais derivados. A motivação principal na utilização deste tipo de transformada baseia-se não só no fato de que esta decomposição apresenta propriedades discriminativas para caracterização de textura [42], mas também, em comparação com outros métodos de decomposição, os coeficientes de características são menos sensíveis a distorções geométricas tais como rotação e escala. Assim, considerando-se que as imagens de textura podem apresentar tais tranformações geométricas, torna-se necessária a existência de um método de extração de características invariante a estas transformações. Neste sentido, o descritor proposto considera a *orientação dominante* ou a *escala dominante* nas texturas para dessa forma alinhar os elementos dos vetores de características obtidos através da média e do desvio padrão de cada uma das sub-bandas decompostas.

Resultados experimentais conduzidos em bases de imagens geradas a partir do banco de imagens Brodatz mostram a superioridade deste descritor no que diz respeito à eficácia da recuperação com relação a outros descritores de textura, tais como Decomposição Piramidal *Steerable* convencional [124] e as *Gabor Wavelets* [87].

1.4.2 *Rotation-invariant Texture Recognition*

O Capítulo 3 (*Rotation-invariant Texture Recognition*) inclui os resultados apresentados em [96].

Este trabalho apresenta os resultados referentes à proposta de um novo sistema de reconhecimento de texturas, em que as imagens podem ser caracterizadas pela presença de rotações. As duas componentes principais do sistema proposto consistem num descritor de texturas invariante à rotação e um método de reconhecimento multi-classe supervisionado baseado no Caminho Ótimo de Florestas (*Optimum Path Forest* [108]).

O descritor utiliza as propriedades discriminativas da Decomposição Piramidal *Steerable* para caracterização de texturas. Para obter invariância à rotação, a orientação dominante nas imagens de textura é encontrada de tal forma que os elementos dos vetores são alinhados em relação a esta orientação.

Por outro lado, o sistema empregou o classificador de padrões baseado no Caminho Ótimo de Floresta [108], o qual encontra os protótipos com zero erro de classificação nos conjuntos de treinamento e, durante a aprendizagem, corrige erros nos conjuntos de avaliação. Combinando as propriedades discriminativas do descritor e classificador, o sistema utiliza vetores de características de baixa dimensão para caracterizar as imagens de textura sem comprometer as taxas de classificação.

Para avaliar a acurácia do sistema, diversos experimentos foram conduzidos em bases de imagens geradas a partir do banco de imagens de Brodatz e os resultados obtidos foram comparados com outras abordagens.

1.4.3 *Learning How to Extract Rotation-Invariant and Scale-Invariant Features from Texture Images*

O Capítulo 4 (*Learning How to Extract Rotation-Invariant and Scale-Invariant Features from Texture Images*) inclui os resultados apresentados em [97].

O objetivo da pesquisa apresentada neste trabalho consiste em combinar resultados obtidos nas áreas de análise de imagem e reconhecimento de padrões para prover mecanismos de aprendizado e reconhecimento automatizado de imagens de textura. Uma importante motivação neste estudo consiste, por um lado, em facilitar aplicações de reconhecimento de textura em que o espaço de armazenamento físico e a eficiência podem ser fatores críticos e, por outro lado, em caracterizar as imagens de textura considerando as suas características naturais, isto é, na presença de distorções geométricas (rotações e escalas).

Para tratar estes desafios no contexto de extração de características utilizamos o descritor proposto baseado na Decomposição Piramidal *Steerable* de modo que a informação relevante de textura é extraída em vetores de características de baixa dimensão, incluindo propriedades invariantes à rotação e à escala. O classificador de padrões baseado no Caminho Ótimo de Floresta é utilizado na etapa de reconhecimento de características do sistema.

Dos resultados obtidos em diversas bases de imagens pode-se demontrar a acurácia do método de reconhecimento proposto. Além disso, diversas comparações foram realizadas com outras abordagens.

1.4.4 *Wavelet-based Fingerprint Image Retrieval*

O Capítulo 5 (*Wavelet-based Fingerprint Image Retrieval*) inclui os resultados apresentados em [93–95, 148].

Neste trabalho apresenta-se uma proposta para recuperação de impressões digitais visando à identificação de pessoas. Para tanto, diferentes etapas relacionadas com a recuperação de imagens por conteúdo, tais como extração de características e avaliações de similaridade, são aqui consideradas.

Alguns tipos de Wavelets para representação e descrição da informação textural em impressões digitais são utilizados. Neste caso, os vetores de características são obtidos considerando-se a média e o desvio padrão das imagens decompostas no domínio das *Wavelets*. Os tipos diferentes de *Wavelets* utilizados neste estudo incluem: *Gabor Wavelets*, Decomposição Estruturada de Árvores *Wavelets* (Tree-Structured Wavelet) utilizando-se Bancos de Filtros Ortogonais e Bi-Ortogonais bem como a Decomposição Piramidal *Steerable*.

Para avaliar a acurácia da recuperação do método proposto, um total de oito diferentes bases foram consideradas. Experimentos foram conduzidos visando avaliar diferentes combinações de *Wavelets* e medidas de similaridade. Os resultados mostram, entre outros, que as Wavelets podem ser utilizadas para caracterizar o núcleo das impressões digitais de modo eficaz.

Capítulo 2

Rotation-Invariant and Scale-Invariant Steerable Pyramid Decomposition for Texture Image Retrieval

2.1 Introduction

Texture-based image retrieval has been an active research topic over the last years. Although several methods have achieved high retrieval rates [30, 53, 67], some of them were evaluated under controlled scenarios, where there is a lack of image distortions, such as rotations and scales. Furthermore, methods that achieved rotation-invariant texture characterizations, where mostly proposed for classification applications. In general, these methods take advantage of a *priori* knowledge about the texture patterns, so that their classification rates can be improved [70, 88, 121].

In this context, the next challenge in texture image retrieval applications consists therefore in achieving rotation-, and scale-invariant feature representations for *non-controlled* environments.

This paper addresses these problems by proposing a new texture descriptor based on Steerable Pyramids. Roughly speaking, a Steerable Pyramid is a method in which images are decomposed into a set of multi-scale, and multi-orientation image subbands, where the basis functions are directional derivative operators [37]. Our motivation in using Steerable Pyramids relies not only on the fact that they have demonstrated discriminability properties for texture characterization [42], but also that unlike other image decomposition methods, the feature coefficients are less modified under the presence of image rotations, or even scales. The proposed descriptor computes the mean and standard deviation of the decomposed image subbands. To obtain rotation or scale invariance, extracted feature vectors are aligned by considering either the **dominant orientation** or **dominant scale** of the input textures.

The main contributions of this paper are: (1) the proposal of a new texture feature representation based on Steerable Pyramid Decomposition to facilitate rotation-invariant, and scale-invariant applications, and (2) a comprehensive evaluation that demonstrates the discriminating power of our approach in characterizing different classes of textures under the presence of image distortions (scales and orientations).

The outline of this paper is as follows. In the next section, we briefly review the fundamentals of the Steerable Pyramid Decomposition. Section 2.3 describes how texture images are characterized to obtain rotation-invariant, and scale-invariant representations. The experimental setup in our study is presented in section 2.4. In section 2.5, experimental results on several datasets are given, and are used to demonstrate the retrieval effectiveness improvement of our approach. Finally, some conclusions are drawn in section 2.6.

2.2 Steerable Pyramid Decomposition

The Steerable Pyramid Decomposition is a linear multi-resolution image decomposition method, by which an image is subdivided into a collection of subbands localized at different scales and orientations [37]. Using a high-, and low-pass filter (H_0, L_0) the input image is initially decomposed into two subbands: a high-, and a low-pass subband, respectively. Further, the low-pass subband is decomposed into K-oriented band-pass portions $B_0, \ldots, B_{K-1}$, and into a lowpass subband L_1. The decomposition is done recursively by subsampling the lower low-pass subband (L_S) by a factor of 2 along the rows and columns. Each recursive step captures different directional information at a given scale. Considering the polar-separability of the filters in the Fourier domain, the first low-, and high-pass filters, are defined as [115]:

$$\begin{array}{rl} L_0(r,\theta) = & L\left(\frac{r}{2},\theta\right)/2 \\ H_0(r,\theta) = & H\left(\frac{r}{2},\theta\right) \end{array} \tag{2.1}$$

where r, θ are the polar frequency coordinates. L, H are raised cosine low-, and high-pass transfer functions:

$$L(r,\theta) = \begin{cases} 2 & r \leq \frac{\pi}{4} \\ 2\cos\left(\frac{\pi}{2}\log_2\left(\frac{4r}{\pi}\right)\right) & \frac{\pi}{4} < r < \frac{\pi}{2} \\ 0 & r \geq \frac{\pi}{2} \end{cases} \tag{2.2}$$

$$B_k(r,\theta) = H(r)G_k(\theta), \qquad k \in [0, K-1] \tag{2.3}$$

$B_k(r,\theta)$ represents the K directional bandpass filters used in the iterative stages, with radial and angular parts, defined as:

$$H(r,\theta) = \begin{cases} 1 & r \geq \frac{\pi}{4} \\ \cos\left(\frac{\pi}{2}\log_2\left(\frac{2r}{\pi}\right)\right) & \frac{\pi}{4} < r < \frac{\pi}{2} \\ 0 & r \leq \frac{\pi}{2} \end{cases} \tag{2.4}$$

$$G_k(\theta) = \begin{cases} \alpha_K \left(\cos\left(\theta - \frac{\pi k}{K}\right)\right)^{K-1} & \left|\theta - \frac{\pi k}{K}\right| < \frac{\pi}{2} \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

where $\alpha_k = 2^{(k-1)} \frac{(K-1)!}{\sqrt{K[2(K-1)]!}}$.
A first level image decomposition using Steerable Pyramid is shown in Figure 2.1.

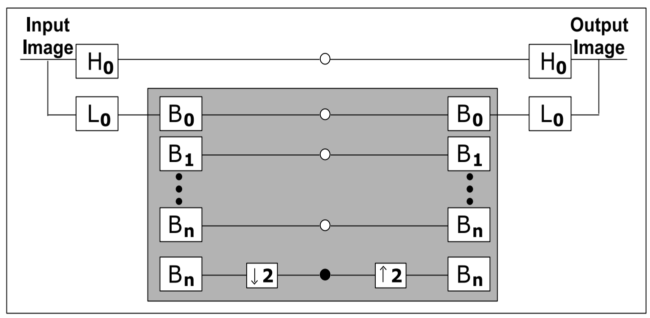


Figura 2.1: First level image decomposition based on Steerable Pyramid Decomposition.

2.3 Texture feature representation

This section describes the proposed modification of Steerable Pyramid Decomposition to obtain the rotation-invariant, and scale-invariant representations, which are used to characterize the texture images.

2.3.1 Texture representation

Roughly speaking, texture images can be seen as a set of basic repetitive primitives characterized by their spatial homogeneity [28]. By applying statistical measures, this information is extracted, and used to capture the relevant image content into feature vectors. More precisely, we use the mean (μ_{mn}) and standard deviation (σ_{mn}) of the energy distribution of the filtered images (S_{mn}), by considering the presence of homogeneous regions in texture images. Given an image $I(x,y)$, its Steerable Pyramid Decomposition is defined as:

$$S_{mn}(x,y) = \sum_{x_1}\sum_{y_1} I(x_1,y_1)B_{mn}(x-x_1,y-y_1) \tag{2.6}$$

where B_{mn} denotes the directional bandpass filters at stage $m = 0,1,\ldots,S-1$, and orientation $n = 0,1,\ldots,K-1$. The energy distribution $(E(m,n))$ of the filtered images at scale m, and at orientation n is defined as:

$$E(m,n) = \sum_{x}\sum_{y} |S_{mn}(x,y)| \tag{2.7}$$

Additionally, the mean (μ_{mn}) and standard deviation (σ_{mn}) of the energy distributions are found as follows:

$$\mu_{mn} = \frac{1}{MN} E_{mn}(x,y) \tag{2.8}$$

$$\sigma_{mn} = \sqrt{\frac{1}{MN}\sum_{x}\sum_{y}\left(|S_{mn}(x,y)| - \mu_{mn}\right)^2} \tag{2.9}$$

The corresponding feature vector $(\vec{f})$ is defined by using the mean and standard deviation as feature elements. It is denoted as:

$$\vec{f} = [\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \ldots, \mu_{S-1K-1}, \sigma_{S-1K-1}] \tag{2.10}$$

2.3.2 Rotation-invariant representation

Rotation-invariant representation is achieved by computing the dominant orientation of the texture images followed by feature alignment. The **dominant orientation** (DO) is defined as the orientation with the highest total energy across the different scales considered during image decomposition [3]. It is computed by finding the highest accumulated energy for the K different orientations considered during image decomposition:

$$DO_i = max\left\{E_0^{(R)}, E_1^{(R)}, \ldots, E_{K-1}^{(R)}\right\} \tag{2.11}$$

where i is the index where the dominant orientation appeared, and:

$$E_n^{(R)} = \sum_{m=0}^{S-1} E(m,n), \quad n = 0,1,\ldots,K-1. \tag{2.12}$$

Note that each $E_n^{(R)}$ covers a set of filtered images at different scales but at same orientation.

Finally, rotation-invariance is obtained by shifting circularly feature elements within the same scales, so that first elements at each scale correspond to dominant orientations.

Let $\vec{f}$ be a feature vector obtained by using a Pyramid Decomposition with $S = 2$ scales, and $K = 3$ orientations:

$$\begin{aligned}\vec{f} = \; & [\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \mu_{02}, \sigma_{02}; \\ & \mu_{10}, \sigma_{10}, \mu_{11}, \sigma_{11}, \mu_{12}, \sigma_{12}]\end{aligned} \tag{2.13}$$

Now suppose that the dominant orientation appears at index $i = 1$ ($DO_{i=1}$), thus the rotation-invariant feature vector, after feature alignment, is represented as follows:

$$\begin{aligned}\vec{f}^R = \; & [\mu_{01}, \sigma_{01}, \mu_{02}, \sigma_{02}, \mu_{00}, \sigma_{00}; \\ & \mu_{11}, \sigma_{11}, \mu_{12}, \sigma_{12}, \mu_{10}, \sigma_{10}]\end{aligned} \tag{2.14}$$

2.3.3 Scale-invariant representation

Similarly, scale-invariant representation is achieved by finding the scale with the highest total energy across the different orientations (**dominant scale**). For this purpose,the dominant scale (DS) at index i is computed as follows:

$$DS_i = max\left\{E_0^{(S)}, E_1^{(S)}, \ldots, E_{S-1}^{(S)}\right\} \tag{2.15}$$

where $E_m^{(S)}$ denotes the accumulated energies across the S different scales:

$$E_m^{(S)} = \sum_{n=0}^{K-1} E(m,n), \quad m = 0, 1, \ldots, S-1. \tag{2.16}$$

Note that each $E_m^{(S)}$ covers a set of filtered images at different orientations for each scale. Let $\vec{f}$ be, again, the feature vector obtained by using a Pyramid Decomposition with $S = 2$ scales, and $K = 3$ orientations:

$$\begin{aligned}\vec{f} = \; & [\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \mu_{02}, \sigma_{02}; \\ & \mu_{10}, \sigma_{10}, \mu_{11}, \sigma_{11}, \mu_{12}, \sigma_{12}]\end{aligned} \tag{2.17}$$

By supposing that the dominant scale was found at index $i = 2$ (second scale in the image decomposition), its scale-invariant version, after feature alignment, is defined as:

$$\begin{aligned}\vec{f}^S = \; & [\mu_{10}, \sigma_{10}, \mu_{11}, \sigma_{11}, \mu_{12}, \sigma_{12}; \\ & \mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \mu_{02}, \sigma_{02}]\end{aligned} \tag{2.18}$$

For both rotation-invariant, and scale-invariant representations, the feature alignment is based on the assumption that to compare similarity between texture images, they should be aligned according their dominant orientations/scales.

2.3.4 Similarity Measure

Similarity between images is obtained by computing the distance of their corresponding feature vectors. The smaller the distance, the more similar the images. Given the query image (i), and the target image (j) in the dataset, the distance between the two patterns is defined as [87]:

$$d(i,j) = \sum_m \sum_n d_{mn}(i,j) \tag{2.19}$$

where:

$$d_{mn}(i,j) = \left| \frac{\mu^i_{mn} - \mu^j_{mn}}{\alpha(\mu_{mn})} \right| + \left| \frac{\sigma^i_{mn} - \sigma^j_{mn}}{\alpha(\sigma_{mn})} \right|, \tag{2.20}$$

$\alpha(\mu_{mn})$ and $\alpha(\sigma_{mn})$ denote the standard deviations of the respective features over the entire dataset. They are used for feature normalization purposes.

2.4 Experimental setup

2.4.1 Datasets

To evaluate the effectiveness of our approach, we selected thirteen texture images obtained from the standard Brodatz dataset. Before being digitized, each of the 512×512 texture images was rotated at different degrees [103]. Figure 2.2 displays the non-rotated version of each of the texture images.

To test the rotation-invariance, and scale-invariance of the method, three different image datasets were generated: non-distorted, rotated, and scaled. The *non-distorted* image dataset was constructed just from input textures with no rotation and scale changes. Each texture image was partitioned into sixteen 128×128 non-overlapping subimages. Thus, this dataset comprises 208 (13×16) different images. The second image dataset is referred to as *rotated* image dataset, and was generated by selecting the four 128×128 innermost subimages from texture images at $0, 30, 60$, and 120 degrees. A total number of 208 images were generated ($13 \times 4 \times 4$). Finally, in the *scaled* image dataset, the 512×512 non-rotated textures were first partitioned into four 256×256 non-overlapping subimages. Each partitioned subimage was further scaled by using four different factors, ranging from 0.6 to 0.9 with 0.1 interval. This led to 208 ($13 \times 4 \times 4$) scaled images.

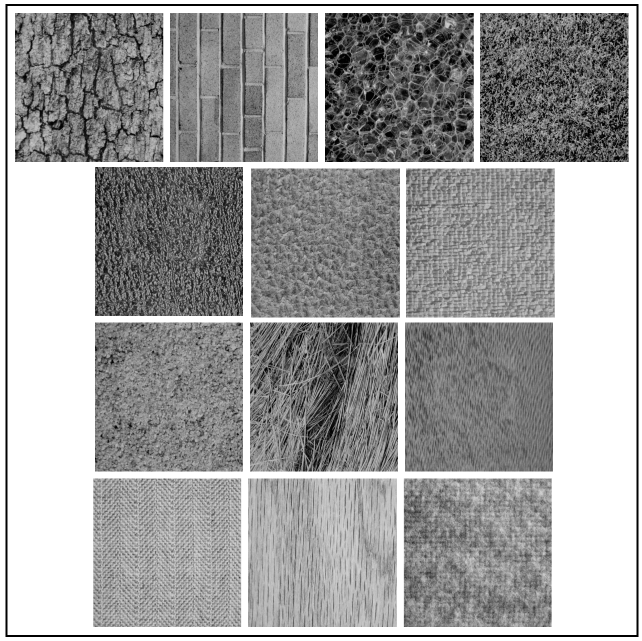


Figura 2.2: Texture images from the Brodatz dataset used in our experiments. From left to right, and from top to bottom, they include: Bark, Brick, Bubbles, Grass, Leather, Pigskin, Raffia, Sand, Straw, Water, Weave, Wood, and Wool.

2.4.2 Retrieval Effectiveness Evaluation

In our experiments, a simulated query is represented by any of the 208 images in a dataset. The relevant images for each query are defined as the 15 remaining subimages from the same input texture. In this context, a total number of 43056 (207×208) queries were performed in each dataset.

The retrieval effectiveness was measured in terms of relevant retrieval average rate, i.e., the percentage of relevant images among the top N retrieved images.

2.5 Experimental Results

Three series of experiments were conducted to evaluate the retrieval effectiveness our method. In the first ones (Section 2.5.1), we evaluate the discriminating power of the conventional Steerable Pyramid Decomposition [124] in characterizing texture images, and how its retrieval effectiveness is affected by the presence of scaled and rotated versions of texture patterns. The second and third series of experiments are used to evaluate the rotation-, and scale-invariant properties of our approach (Sections 2.5.2, 2.5.3, respectively). Comparisons with the conventional Pyramid Decomposition [124], and with a recent proposal for rotation, and scale-invariance texture retrieval based on Gabor Wavelets [46] are further discussed.

We used a Steerable Pyramid having four orientations ($K = 4$), and two levels of decomposition ($S = 2$), leading thus to a feature vector of 16 elements ($4 \times 2 \times 2$). Our experiments agree with [30] in that the most relevant textural information is contained in those two levels, since little retrieval improvement is achieved by varying the number of scales from two to three levels during image decomposition. Furthermore, our motivations in using small size feature vectors are: (1) to show that the retrieval effectiveness of our approach is not compromised, and (2) to facilitate image retrieval applications where data storage capacity is a limitation.

2.5.1 Effectiveness of conventional Steerable Pyramid Feature Representation

The retrieval effectiveness of the conventional Steerable Pyramid is shown in Figure 2.3. It compares, for each class of texture, the average rate of retrieving the relevant images for the *non-distorted* image dataset, the *rotated* image dataset *without* rotation-invariant representation, and the *scaled* image dataset *without* scale-invariant representation.

In the case of the *non-distorted* dataset, it can be noticed that the Steerable Pyramid presents good retrieval accuracy for almost all classes. However, the effectiveness is low for textures that have, either no strong direction (Bark, Bubbles), or for textures whose subimages present visual dissimilarities among each other (Brick). Additionally, we can see that for the *rotated* image dataset, the retrieval effectiveness decays rapidly for images with a defined direction (Brick, Leather, Pigskin, Raffia, Straw, Water, Wood, Wool). This happens because feature coefficients of self-similar subimages are rotated (shifted). Moreover, note that, in the *scaled* image dataset, the retrieval effectiveness is less affected than in the case of the *rotated* image dataset. However, images whose fine patterns become more visible, when increasing the scale factor, are most affected (Leather, Raffia, Sand, and Weave). In this case, the "zoomed" micro-patterns produce enough discriminatory information, so that images belonging to the same texture class cannot be judged as being similar.

2.5.2 Effectiveness of Rotation Invariance Representation

To show both the improvement over the conventional Steerable Pyramid, and the capability of our method for characterizing rotated texture images, we used the *rotated* image dataset. From Figure 2.4, we can see that for almost all classes the retrieval accuracy was

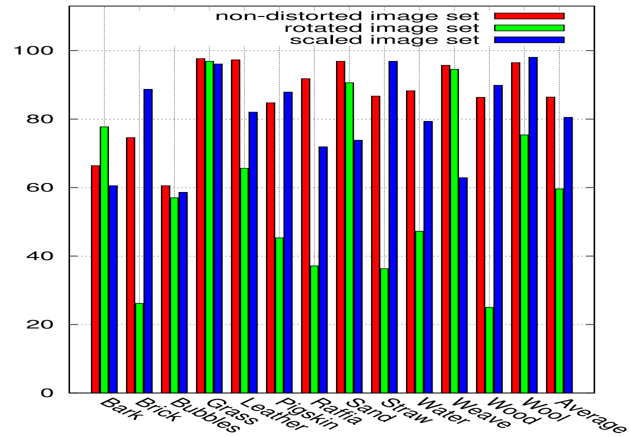


Figura 2.3: Average retrieval rates for *non-distorted* image dataset, *rotated* image dataset *without* rotation-invariant representation, and the *scaled* image dataset *without* scale-invariant representation using the conventional Steerable Pyramid Decomposition having $S = 2$ scales and $K = 4$ orientations.

dramatically increased. This observation is more notorious in images having strong direction (Brick, Leather, Pigskin, Raffia, Straw, Water, Wood), since our feature alignment considers the dominant orientation in the images. Further, Figure 2.5 shows that our approach outperforms the recent proposal for rotation-invariant characterization using based on Gabor Wavelets [46]. However, the latter method presents better retrieval accuracy for images that have an uniform scattered region, i.e, some unique primitives appear at any location in the textures, including, for example, image borders (Water, Wood). In this case, Gabor Wavelets perform better, since at least one filter in the filterbank covers that subregion.

2.5.3 Effectiveness of Scale Invariance Representation

In this subsection, our approach is compared with the conventional Steerable Pyramid Decomposition, and Gabor Wavelets for the *scaled* image dataset (Figures 2.6, 2.7). From Figure 2.6, we can notice that the highest retrieval accuracy improvement is achieved in textures characterized by the presence of micro-patterns (Pigskin, Sand, Weave). By using different levels of scale decompositions, these micro-patterns are highlighted, and therefore are more distinctive even for textures belonging to the same texture class. Furthermore, feature vectors are aligned according to the dominant scale in texture images, improving the effectiveness of the retrieval process.

In Figure 2.7, we compare the retrieval effectiveness of our approach with the one

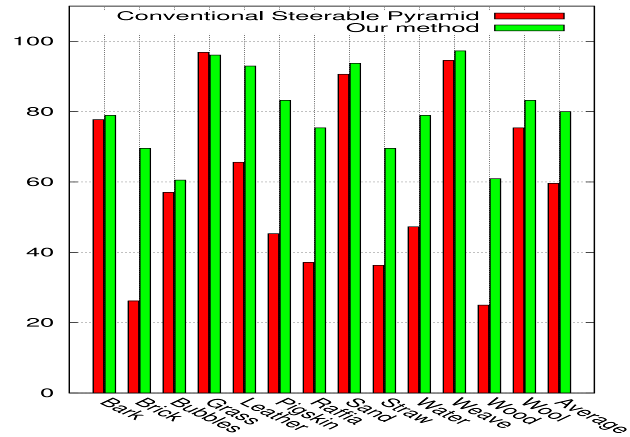


Figura 2.4: Average retrieval rates per texture class for *rotated* image dataset *without* and *with* rotation-invariant representation, respectively.

using the modified Gabor Wavelets. Our method yields better retrieval effectiveness for all classes except for the Leather texture pattern.

Note that more discriminatory information is obtained by downsampling rows and columns (Steerable Pyramid approach) than by using filterbanks at few scales (Gabor Wavelets approach).

2.5.4 Results Summarization

A summary of our experimental results is provided in Table 2.1. It compares the average rates of retrieving the 15 most similar images for each of the input textures in the *rotated*, and *scaled* image datasets. We can see that our method (M-SPyd) improves retrieval rates over the conventional Steerable Pyramid (C-SPyd) and Gabor Wavelets (GWs) on both image datasets. Note that the retrieval effectiveness was improved in almost 20% for the rotated image dataset, whereas for the scaled image dataset it was improved from 80.5% to 86.51% over the conventional Steerable Pyramid.

2.5.5 Image Retrieval Examples

Figure 2.8 displays four retrieval examples of the three evaluated methods. The two first queries were selected from the rotated-image datasets. They correspond to the classes for which our method and the Gabor Wavelets achieved the highest retrieval accuracy (Raffia and Water, respectively). Similarly, the remaining two image queries were selected to illustrate the scale-invariance capabilities. They belong to the Sand and Leather classes.

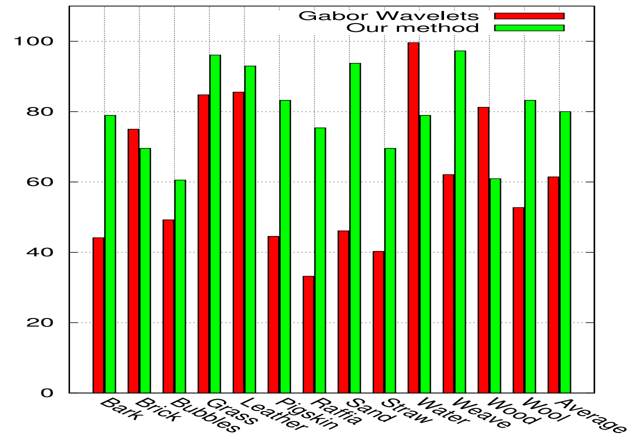


Figura 2.5: Comparison of rotation-invariant retrieval performances for the *rotated* image dataset using the recent Gabor-based approach [46], and our approach.

	M-SPyd	C-SPyd	GWs
Rotated image dataset	**80.02%**	59.62%	61.42%
Scaled image dataset	**86.51%**	80.5%	60.7%
Feature dimensionality	16	16	16

Tabela 2.1: Average retrieval rates and feature dimensionalities for the methods used in our experiments.

The input queries are shown on the first columns, and their relevant retrieved images are displayed in increasing order on the following columns. Furthermore, for each query image, the retrieved relevant images shown on the first, second, and third rows, are obtained by considering respectively the conventional Steerable Pyramid, our approach, and Gabor Wavelets.

From the retrieved images, we can see that our method outperforms the conventional Steerable Pyramid for all query images. Additionally, for the cases where Gabor Wavelets achieved highest retrieval accuracy (second, and fourth queries), our method retrieved almost all relevant images.

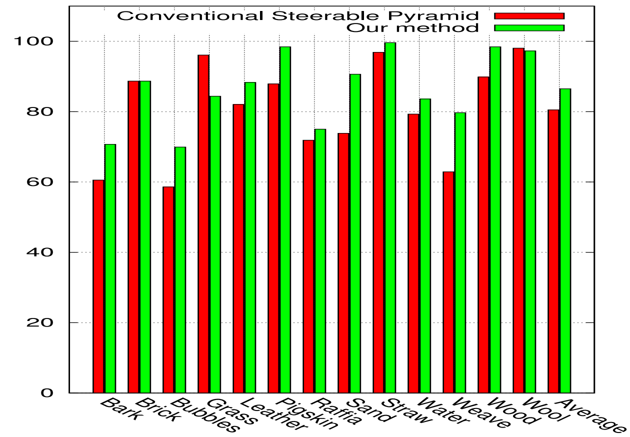


Figura 2.6: Average retrieval rates per texture class for *scaled* image dataset *without* and *with* rotation-invariant representation, respectively.

2.6 Conclusions

In this paper, a new method for facilitating rotation-invariant, and scale-invariant image retrieval applications was introduced. Our approach exploits the discriminability properties of the Steerable Pyramid Decomposition for texture characterization, and by taking into account either the **dominant orientation** or **dominant scale** of the input textures, a new feature descriptor is proposed.

We performed a total number of 43056 different queries in three image datasets. Experimental results obtained in those datasets are very encouraging, since the new method demonstrated retrieval rates improvements for both rotated, and scaled image datasets. More specifically, the rotation-invariant and scale-invariant average retrieval rates of our method are increased respectively from 59.62% to 80.02% and from 80.5% to 86.51% over the conventional Steerable Pyramid. It is worth mentioning, that the average retrieval rates of Gabor Wavelets presented less retrieval accuracy being 61.42% and 60.7% for the rotated and scaled image datasets, respectively.

Future work on this matter will include extending this method for both rotated and scaled image characterization. Furthermore, we plan to use this approach for problems concerning texture segmentation and recognition.

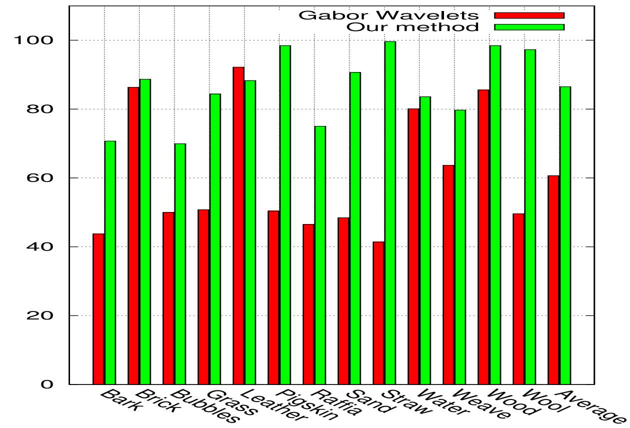


Figura 2.7: Comparison of scale-invariant retrieval performances for the *scaled* image dataset using the recent Gabor-based approach [46], and our approach.

2.7 Acknowledgments

Javier Montoya would like to thank professor Serge Belongie for his valuable discussions and suggestions, and is also grateful to CNPq for the financial support (134990/2005-6). This work is partially supported by Microsoft Escience and Tablet PC Technology and Higher Education projects, CNPq (Proc. 477039/2006-5 and 311309/2006-2), FAPESP, CAPES, and FAEPEX.

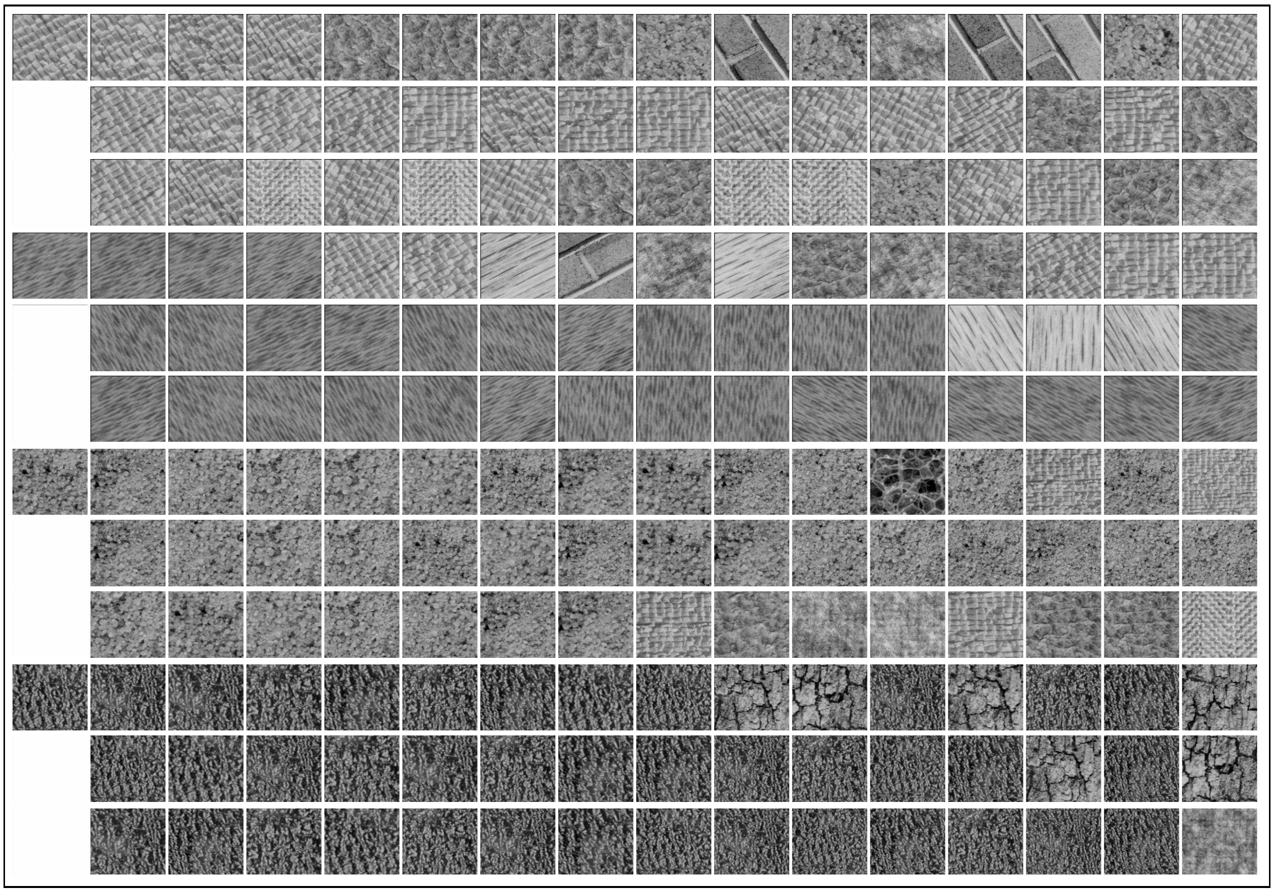


Figura 2.8: Image retrieval examples for each of the three methods: conventional Steerable Pyramid, our approach, and Gabor Wavelets (first, second, and third rows, respectively). Each query image is shown on the first column, and its 15 top matches are displayed in increasing order.

Capítulo 3

Rotation-invariant Texture Recognition

3.1 Introduction

In the last years, several image recognition systems have been proposed in the literature as a result of many research efforts [11, 50]. Although those approaches have achieved high classification rates, most of them have not been widely evaluated in texture image databases. Traditionally, texture images may be characterized by: (1) small inter-class variations, i.e, textures belonging to different classes may appear quite similar, especially in terms of their global patterns (coarseness, smoothness, etc.), and (2) the presence of image distortions such as rotations. In this sense, texture pattern recognition is a still open task. The next challenge in texture classification should be, therefore, to achieve rotation-invariant feature representations for *non-controlled* environments. To address some of these limitations, this work proposes a new texture classification method, which is characterized by: (1) a new texture image descriptor based on Steerable Pyramid Decomposition, which encodes the relevant texture information in small size feature vectors including rotation-invariant characterization, and (2) a novel multi-class object recognition method based on the Optimum Path Forest classifier [108].

Roughly speaking, a Steerable Pyramid is a method by which images are decomposed into a set of multi-scale, and multi-orientation image subbands, where the basis functions are directional derivative operators [37]. Our motivation in using Steerable Pyramids relies on that, unlike other image decomposition methods, the feature coefficients are less affected by image distortions. Furthermore, the Optimum Path Forest Classifier is a recent approach that handles non separable classes, without the necessity of using boosting procedures to increase its performance, resulting thus in a faster and more accurate classifier for object recognition. By combining the discriminating power of our image descriptor and classifier, our system uses small size feature vectors to characterize texture images without compromising overall classification rates. In this way, texture classification applications, where data storage capacity is a limitation, are further facilitated.

The outline of this paper is as follows. In the next section, we briefly review the fundamentals of the Steerable Pyramid Decomposition. Section 3.3 describes how texture images are characterized to obtain rotation-invariant representations. Section 3.4 introduces the Optimum Path Forest classifier method. The experimental setup conducted in our study is presented in Section 3.5. In section 3.6, experimental results on several datasets are given and are used to demonstrate the recognition accuracy improvement of our approach. Comparisons with other texture feature representations and classifiers are further discussed. Finally, some conclusions are drawn in Section 3.7.

3.2 Steerable Pyramid Decomposition

The Steerable Pyramid Decomposition is a linear multi-resolution image decomposition method, by which an image is subdivided into a collection of subbands localized at different scales and orientations [37]. Using a high-, and low-pass filter (H_0, L_0) the input image is initially decomposed into two subbands: a high-, and a low-pass subband, respectively. Further, the low-pass subband is decomposed into K-oriented band-pass portions $B_0, \ldots, B_{K-1}$, and into a lowpass subband L_1. The decomposition is done recursively by subsampling the lower low-pass subband (L_S) by a factor of 2 along the rows and columns. Each recursive step captures different directional information at a given scale. Considering the polar-separability of the filters in the Fourier domain, the first low-, and high-pass filters, are defined as [115]:

$$L_0(r,\theta) = L\left(\frac{r}{2},\theta\right)/2 \qquad H_0(r,\theta) = H\left(\frac{r}{2},\theta\right) \tag{3.1}$$

where r, θ are the polar frequency coordinates. L, H are raised cosine low-, and high-pass transfer functions:

$$L(r,\theta) = \begin{cases} 2 & r \leq \frac{\pi}{4} \\ 2\cos\left(\frac{\pi}{2}\log_2\left(\frac{4r}{\pi}\right)\right) & \frac{\pi}{4} < r < \frac{\pi}{2} \\ 0 & r \geq \frac{\pi}{2} \end{cases} \tag{3.2}$$

$$B_k(r,\theta) = H(r)G_k(\theta), \qquad k \in [0, K-1] \tag{3.3}$$

$B_k(r,\theta)$ represents the K directional bandpass filters used in the iterative stages, with radial and angular parts, defined as:

$$H(r,\theta) = \begin{cases} 1 & r \geq \frac{\pi}{4} \\ \cos\left(\frac{\pi}{2}\log_2\left(\frac{2r}{\pi}\right)\right) & \frac{\pi}{4} < r < \frac{\pi}{2} \\ 0 & r \leq \frac{\pi}{2} \end{cases} \tag{3.4}$$

$$G_k(\theta) = \begin{cases} \alpha_K\left(\cos\left(\theta - \frac{\pi k}{K}\right)\right)^{K-1} & \left|\theta - \frac{\pi k}{K}\right| < \frac{\pi}{2} \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

where $\alpha_k = 2^{(k-1)}\frac{(K-1)!}{\sqrt{K[2(K-1)]!}}$.

3.3 Texture feature representation

This section describes the proposed modification of Steerable Pyramid Decomposition to obtain rotation-invariant representations, used to characterize the texture images.

3.3.1 Texture representation

Roughly speaking, texture images can be seen as a set of basic repetitive primitives characterized by their spatial homogeneity [28]. By applying statistical measures, this information is extracted, and used to capture the relevant image content into feature vectors. More precisely, by considering the presence of homogeneous regions in texture images, we use the mean (μ_{mn}) and standard deviation (σ_{mn}) of the energy distribution of the filtered images (S_{mn}). Given an image $I(x,y)$, its Steerable Pyramid Decomposition is defined as:

$$S_{mn}(x,y) = \sum_{x_1}\sum_{y_1} I(x_1,y_1)B_{mn}(x-x_1, y-y_1) \tag{3.6}$$

where B_{mn} denotes the directional bandpass filters at stage $m = 0,1,\ldots,S-1$, and orientation $n = 0,1,\ldots,K-1$. The energy distribution ($E(m,n)$) of the filtered images at scale m, and at orientation n is defined as:

$$E(m,n) = \sum_x\sum_y |S_{mn}(x,y)| \tag{3.7}$$

Additionally, the mean (μ_{mn}) and standard deviation (σ_{mn}) of the energy distributions are found as follows:

$$\mu_{mn} = \frac{1}{MN}E_{mn}(x,y) \qquad \sigma_{mn} = \sqrt{\frac{1}{MN}\sum_x\sum_y\left(|S_{mn}(x,y)| - \mu_{mn}\right)^2} \tag{3.8}$$

The corresponding feature vector ($\vec{f}$) is defined by using the mean and standard deviation as feature elements. It is denoted as:

$$\vec{f} = [\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \ldots, \mu_{S-1K-1}, \sigma_{S-1K-1}] \tag{3.9}$$

3.3.2 Rotation-invariant representation

Rotation-invariant representation is achieved by computing the dominant orientation of the texture images followed by feature alignment. The **dominant orientation** (DO) is defined as the orientation with the highest total energy across the different scales considered during image decomposition. It is computed by finding the highest accumulated energy for the K different orientations considered during image decomposition:

$$DO_i = max\left\{E_0^{(R)}, E_1^{(R)}, \ldots, E_{K-1}^{(R)}\right\} \tag{3.10}$$

where i is the index where the dominant orientation is found, and:

$$E_n^{(R)} = \sum_{m=0}^{S-1} E(m,n), \quad n = 0, 1, \ldots, K-1. \tag{3.11}$$

Note that each $E_n^{(R)}$ covers a set of filtered images at different scales but at same orientation. Finally, rotation-invariance is obtained by shifting circularly feature elements within the same scales, so that first elements at each scale correspond to dominant orientations. This process is based on the assumption that to classify textures, they should be rotated so that their dominant directions are the same. Further, it has been proved that image rotation in spatial domain is equivalent to circular shift of feature vector elements [149].

3.4 Texture feature recognition

This section aims to present the new approach to pattern recognition called OPF (Optimum Path Forest), which has been demonstrated to be generally more efficient than Artificial Neural Networks and Support Vector Machines [108]. The OPF approach works by modeling the patterns as being nodes of a graph in the feature space, where every pair of nodes are connected by an arc (complete graph). This classifier creates a discrete optimal partition of the feature space such that any unknown sample can be classified according to this partition. This partition is an optimum path forest computed in $\Re^n$ by the image foresting transform (IFT) algorithm [33].

Let Z_1, Z_2, and Z_3 be training, evaluation, and test sets with $|Z_1|$, $|Z_2|$, and $|Z_3|$ samples such as feature vectors. Let $\lambda(s)$ be the function that assigns the correct label i,

$i = 1, 2, \ldots, c$, from class i to any sample $s \in Z_1 \cup Z_2 \cup Z_3$. Z_1 and Z_2 are labeled sets used to the design of the classifier and the unseen set Z_3 is used to compute the final accuracy of the classifier. Let $S \subset Z_1$ be a set of prototypes of all classes (i.e., key samples that best represent the classes). Let v be an algorithm which extracts n attributes (texture properties) from any sample $s \in Z_1 \cup Z_2 \cup Z_3$ and returns a vector $\vec{v}(s) \in \Re^n$. The distance $d(s,t)$ between two samples, s and t, is the one between their feature vectors $\vec{v}(s)$ and $\vec{v}(t)$ (e.g., Euclidean or any valid metric).

Let (Z_1, A) be a complete graph whose nodes are the samples in Z_1. We define a path as being a sequence of distinct samples $\pi = \langle s_1, s_2, \ldots, s_k \rangle$, where $(s_i, s_{i+1}) \in A$ for $1 \leq i \leq k-1$. A path is said *trivial* if $\pi = \langle s_1 \rangle$. We assign to each path π a cost $f(\pi)$ given by a path-cost function f. A path π is said optimum if $f(\pi) \leq f(\pi')$ for any other path π', where π and π' end at a same sample s_k. We also denote by $\pi \cdot \langle s, t \rangle$ the concatenation of a path π with terminus at s and an arc (s,t). The OPF algorithm uses the path-cost function f_{max}, because of its theoretical properties for estimating optimum prototypes:

$$\begin{aligned} f_{max}(\langle s \rangle) &= \begin{cases} 0 & \text{if } s \in S, \\ +\infty & \text{otherwise} \end{cases} \\ f_{max}(\pi \cdot \langle s, t \rangle) &= \max\{f_{max}(\pi), d(s,t)\} \end{aligned} \tag{3.12}$$

We can observe that $f_{max}(\pi)$ computes the maximum distance between adjacent samples in π, when π is not a trivial path.

The OPF algorithm assigns one optimum path $P^*(s)$ from S to every sample $s \in Z_1$, forming an optimum path forest P (a function with no cycles which assigns to each $s \in Z_1 \backslash S$ its predecessor $P(s)$ in $P^*(s)$ or a marker *nil* when $s \in S$. Let $R(s) \in S$ be the root of $P^*(s)$ which can be reached from $P(s)$. The OPF algorithm computes for each $s \in Z_1$, the cost $C(s)$ of $P^*(s)$, the label $L(s) = \lambda(R(s))$, and the predecessor $P(s)$, as follows.

Algorithm 1. – OPF ALGORITHM

INPUT:	A λ-labeled training set Z_1, prototypes $S \subset Z_1$ and the pair (v, d) for feature vector and distance computations.
OUTPUT:	Optimum path forest P, cost map C and label map L.
AUXILIARY:	Priority queue Q and cost variable *cst*.

1. *For each* $s \in Z_1 \backslash S$, *set* $C(s) \leftarrow +\infty$.
2. *For each* $s \in S$, *do*
3. $\quad$ └ $C(s) \leftarrow 0$, $P(s) \leftarrow nil$, $L(s) \leftarrow \lambda(s)$, *and insert* s *in* Q.
4. *While* Q *is not empty, do*

5. *Remove from Q a sample s such that $C(s)$ is minimum.*
6. *For each $t \in Z_1$ such that $t \neq s$ and $C(t) > C(s)$, do*
7. *Compute $cst \leftarrow \max\{C(s), d(s,t)\}$.*
8. *If $cst < C(t)$, then*
9. *If $t \in Q$, then remove t from Q.*
10. *$P(t) \leftarrow s$, $L(t) \leftarrow L(s)$, $C(t) \leftarrow cst$, and insert t in Q.*

Lines $1-3$ initialize maps and insert prototypes in Q. The main loop computes an optimum path from S to every sample s in a non-decreasing order of cost (Lines $4-10$). At each iteration, a path of minimum cost $C(s)$ is obtained in P when we remove its last node s from Q (Line 5). Lines $8-10$ evaluate if the path that reaches an adjacent node t through s is cheaper than the current path with terminus t and update the position of t in Q, $C(t)$, $L(t)$ and $P(t)$ accordingly. The label $L(s)$ may be different from $\lambda(s)$, leading to classification errors in Z_1. The training finds prototypes with none classification errors in Z_1. The OPF algorithm works with two phases: training and classification (test), as follows.

3.4.1 Training phase

We say that S^* is an optimum set of prototypes when Algorithm 1 propagates the labels $L(s) = \lambda(s)$ for every $s \in Z_1$. Set S^* can be found by exploiting the theoretical relation between *Minimum Spanning Tree* (MST) [24] and optimum path tree for f_{max}. The training essentially consists of finding S^* and an OPF classifier rooted at S^*.

By computing an MST in the complete graph (Z_1, A), we obtain a connected acyclic graph whose nodes are all samples in Z_1 and the arcs are undirected and weighted by the distance d between the adjacent sample feature vectors. This spanning tree is optimum in the sense that the sum of its arc weights is minimum as compared to any other spanning tree in the complete graph. In the MST, every pair of samples is connected by a single path which is optimum according to f_{max}. That is, for any given sample $s \in Z_1$, it is possible to direct the arcs of the MST such that the result will be an optimum path tree P for f_{max} rooted at s. The optimum prototypes are the closest elements in the MST with different labels in Z_1. By removing the arcs between different classes, their adjacent samples become prototypes in S^* and Algorithm 1 can compute an optimum path forest with none classification errors in Z_1 without data overfitting [25].

3.4.2 Classification

For any sample $t \in Z_3$, the OPF consider all arcs connecting t with samples $s \in Z_1$, as though t were part of the graph. Considering all possible paths from S^* to t, we wish to find the optimum path $P^*(t)$ from S^* and label t with the class $\lambda(R(t))$ of its most strongly connected prototype $R(t) \in S^*$. This path can be identified incrementally, by evaluating the optimum cost $C(t)$ as

$$C(t) = \min\{\max\{C(s), d(s,t)\}\},\ \forall s \in Z_1. \tag{3.13}$$

Let the node $s^* \in Z_1$ be the one that satisfies the above equation (i.e., the predecessor $P(t)$ in the optimum path $P^*(t)$). Given that $L(s^*) = \lambda(R(t))$, the classification simply assigns $L(s^*)$ as the class of t. An error occurs when $L(s^*) \neq \lambda(t)$.

3.4.3 Learning algorithm

The performance of the OPF classifier improves when the closest samples from different classes are included in Z_1, because the method finds prototypes that will work as sentinels in the frontier between classes. The learning algorithm replaces *irrelevant* samples of Z_1 by errors in Z_2, and secondly other samples of Z_2 are replaced by *irrelevant* samples of Z_1. In both cases, the algorithm never let a sample of Z_2 return to Z_1. The learning algorithm essentially tries to identify these prototypes from a few iterations of classification over Z_2.

In order to define irrelevant samples, the OPF algorithm can identify all samples in Z_1 that participated in the classification task of any node $t \in Z_2$. The OPF consider all right and wrong classifications in Z_2. When $t \in Z_2$ is correctly/incorrectly classified, we add one to the number of right/wrong classifications, of every sample $r \in Z_1$ in the optimum path $P^*(t)$ from $R(t) \in S^*$ to s^*. In that way, the learning algorithm outputs the final projected classifier, which can be now used to predict the labels of Z_3.

3.5 Experimental setup

3.5.1 Datasets

To evaluate the accuracy of our approach, we selected thirteen texture images obtained from the standard Brodatz database. Before being digitized, each of the 512×512 texture images was rotated at different degrees [103]. Figure 3.1 displays the non-rotated version of each of the texture images.

From this database, three different image datasets were generated: *non-distorted*, *rotated-set A*, and *rotated-set B*. The *non-distorted* image dataset was constructed just from the original input textures (i.e. texture patterns at 0 degrees). Each texture image was partitioned into sixteen 128×128 non-overlapping subimages. Thus, this dataset comprises 208 (13×16) different images. Furthermore, images belonging to this dataset will be used in the learning stage of our classifier. The second image dataset is referred to

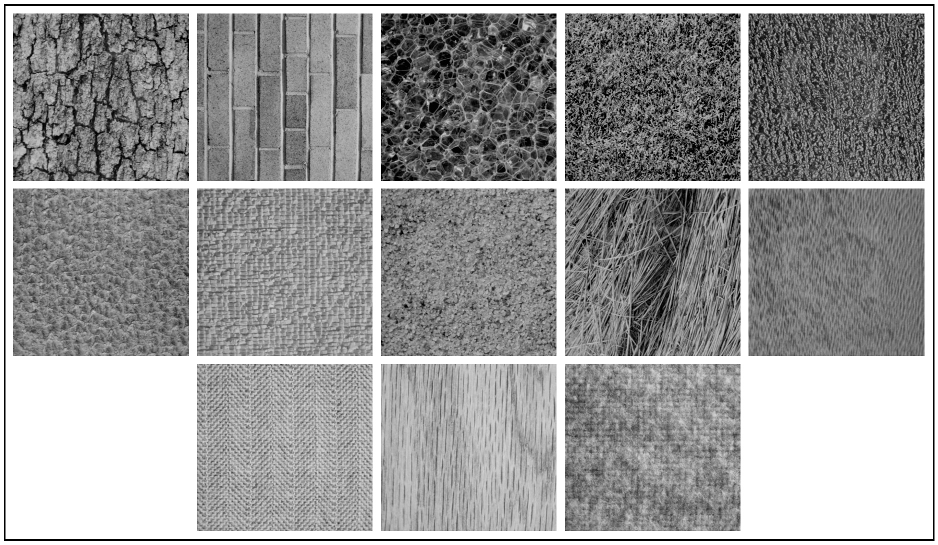


Figura 3.1: Texture images from the Brodatz dataset used in our experiments. From left to right, and from top to bottom, they include: Bark, Brick, Bubbles, Grass, Leather, Pigskin, Raffia, Sand, Straw, Water, Weave, Wood, and Wool.

as *rotated image dataset A*, and was generated by selecting the four 128×128 innermost subimages from texture images at $0, 30, 60$, and 120 degrees. A total number of 208 images were generated ($13 \times 4 \times 4$). Finally, in the *rotated image dataset B*, we selected the four 128×128 innermost subimages of the rotated image textures (512×512) at $0, 30, 60, 90, 120, 150$ and 200 degrees. This led to 364 ($13 \times 4 \times 4$) dataset images. *Rotated image dataset A*, as well as *rotated image dataset B* will be used for recognition purposes.

3.5.2 Classification evaluation

In our experiments, the accuracy was measured by taking into account the possibility of having classes with different cardinalities. Let $N(i)$, $i = 1, 2, \ldots, c$, be the number of samples in each class i and $N = N(1) \cup N(2) \ldots \cup N(c)$ be the whole dataset. We define the partial errors $e_{i,1}$ and $e_{i,2}$ as follows:

$$e_{i,1} = \frac{FP(i)}{|N| - |N(i)|} \quad \text{and} \quad e_{i,2} = \frac{FN(i)}{|N(i)|}, \; i = 1, \ldots, c. \tag{3.14}$$

where $FP(i)$ and $FN(i)$ denote both false positives and false negatives, respectively. That is, $FP(i)$ represents the number of samples of other classes that were classified as being from the class i. In addition, $FN(i)$ represents the number of samples in class i that were misclassified. The errors $e_{i,1}$ and $e_{i,2}$ are then used to define the accumulated partial error of class i as:

$$E(i) = e_{i,1} + e_{i,2}. \tag{3.15}$$

Finally, the resulting classification accuracy $\mathcal{L}$ is found:

$$\mathcal{L} = \frac{2c - \sum_{i=1}^{c} E(i)}{2c} = 1 - \frac{\sum_{i=1}^{c} E(i)}{2c}. \tag{3.16}$$

3.6 Experimental results

To demonstrate the discriminating power of our proposed method for recognizing rotated texture patterns, we conducted two series of experiments. In the first series of experiments (Subsection 3.6.1), we evaluated the effectiveness of the proposed rotation-invariant representation against two other approaches: the conventional Pyramid Decomposition [124] and with a recent proposal based on Gabor Wavelets [3]. Furthermore, the second series of experiments (Subsection 3.6.2), were used to evaluate the recognition accuracy of the novel OPF multi-class classifier. Note that, both series of experiments were conducted using *rotated-sets* A and B. The *rotated-set* A was used to analyze the recognition accuracy of our method under the presence of few rotated versions of texture patterns. In addition, experiments conducted in the *rotated-set* B consider several rotated versions of different texture patterns.

In both series of experiments, we used Steerable Pyramids having different decomposition levels ($S = 2, 3, 4$) at several orientations ($K = 4, 5, 6, 7, 8$). Our experiments agree with [30] in that, the most relevant textural information in images is contained in the first two levels of decomposition, since little recognition improvement is achieved by varying the number of scales during image decomposition. Therefore, we focus our discussions on image decompositions having ($S = 2, 3$) scales. The dimensionality of the feature vectors depends on the number of scales (S) and on the number of orientations (O) considered during image decomposition and it is computed as follows: $2 \times O \times S$. Furthermore, an important motivation in our study was to use small size feature vectors, in order: (1) to show that the recognition accuracy of our approach is not compromised, and (2) to facilitate texture recognition applications where data storage capacity is a limitation.

3.6.1 Effectiveness of the rotation invariance representation

To analyze the texture characterization capabilities of our method against the convetional Pyramid Decomposition [124] and the Gabor Wavelets [3], we used Gaussian-kernel

Support Vector Machines (SVMs) as texture classification mechanisms. Note that, the SVM parameters were optimized by using the cross-validation method [1].

Figure 3.2 compares the recognition accuracy obtained by those three methods in the *rotated-set A*, whereas Figure 3.3 depicts the recognition accuracy obtained in the *rotated-set B*. From both figures, it can be seen that our image descriptor outperforms the other two approaches, regardless of the number of scales or orientations used for extracting the feature vectors.

In the case of the *rotated-set A*, the higher classification accuracies achieved by our method were obtained by using 7 orientations, which corresponds to image rotations in steps of 25.71°. Those accuracies are respectively 100% and 97.31% for two and three decomposition levels ($s = 2, 3$). The corresponing classification accuracies obtained by the Gabor Wavelets are: 90.36% and 93.90% ($s = 2, 3$;$o = 7$), whereas for the conventional Steerable Pyramid those accuracies are: 89.67% and 90.36%. Furthermore, the accuracies using $o = 6, 7, 8$ orientations are very close to each other. Therefore, $s = 2$ and $o = 6$ are the most appropiate parameter combinations for our rotation-invariant image descriptor, and at the same time, low dimensionality feature vectors are obtained.

In the case of the *rotated-set B*, the higher classification accuracies achieved by our descriptor, were again obtained by using 7 orientations. Classification rates of 95.86% and 95.73% correspond respectively to feature vectors with $s = 2, 3$ scales and $o = 7$ orientations. Further, it is found that both Gabor Wavelets and Conventional Steerable Pyramid Decomposition present lower classification rates, being respectively 91.05% and 95.35% for the first method, and 84.22%, 84.23% for the second one. As in the results obtained in *rotated-set A*, we can notice that the achieved classification accuracies are very close to each other, when using $o = 6, 7$ or $o = 8$ orientations. From these results, we can reinforce that the most appropriate parameter settings for our descriptor are $s = 2$ scales and $o = 6$ orientations.

Furthermore, from the bar graphs shown in Figures 3.2 and 3.3, the best performance obtained by the rotation-invariant Gabor method is as good as our descriptor. However, this rate is obtained at $s = 3$ scales, whereas the proposed descriptor achieves the same performance using only $s = 2$ scales. In this sense, an important advantage of our method is its high performance rate at low size feature vectors. It is worth to mention that in our experiments the OPF classifier was at least 10 times faster than the SVM classifier[2].

[1]We used the well known LIBSVM package [19]

[2]Due to the lack of space, we could not present a detailed study of the execution time

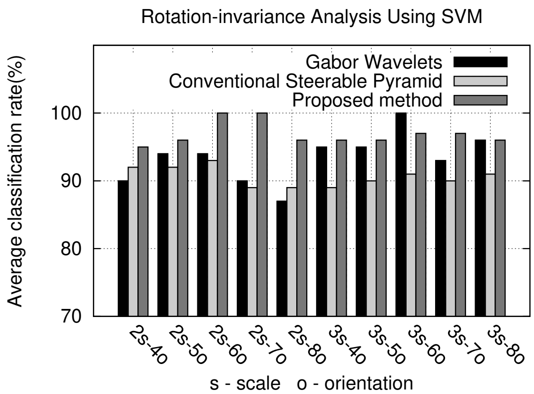


Figura 3.2: Classification accuracy comparison using the SVM classifier obtained in *rotated-set A* using ($S = 2, 3$) scales with ($K = 4, 5, 6, 7, 8$) orientations for Gabor Wavelets, conventional Steerable Pyramid decomposition and our method.

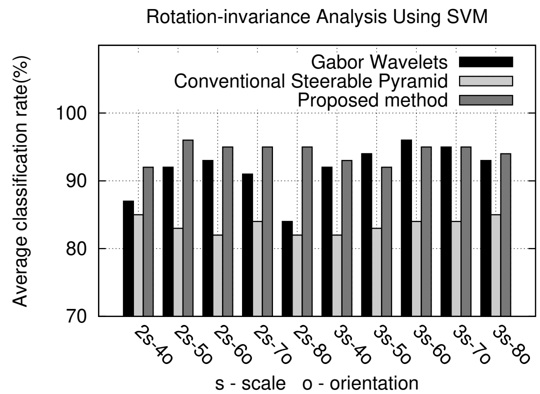


Figura 3.3: Classification accuracy comparison using the SVM classifier obtained in *rotated-set B* using ($S = 2, 3$) scales with ($K = 4, 5, 6, 7, 8$) orientations for Gabor Wavelets, conventional Steerable Pyramid decomposition and our method.

3.6.2 Effectiveness of the multiclass recognition method

In the previous subsection, we showed that our proposed rotation-invariant image descriptor outperforms the other two methods. Therefore, our objective now is to show the recognition improvement of our classifier over the SVM approach. It can be seen from Figure 3.4, that for almost all feature extraction configurations, the recognition rates of the OPF classifier are higher than those of the SVM classifier. However, the latter method presents the same recognition rates as the OPF classifier when using $s = 2$ scales and $o = 6, 7$ orientations. In the case of the image *rotated-set B*, our classifier yields better recognition rates for all feature extraction configurations (See Figure 3.5). By considering that it was found, that the most appropriate parameter settings for our descriptor are $s = 2$ scales and $o = 6$ orientations, it is worth to mention, that by using this configuration, the recognition accuracy obtained by the OPF classifier is 98.49% in comparison with the corresponding accuracy of 95.48% obtained by the SVM classifier.

3.6.3 Results Summarization

A summary of our experimental results is provided in Tables 3.1, and 3.2. Table 3.1 compares for each dataset, the mean recognition rates obtained by the three texture image descriptors using different scales ($s = 2, 3$) and different orientations ($o = 4, 5, 6, 7, 8$). In this set of experiments, we used Gaussian-kernel Support Vector Machines (SVMs) as texture classification mechanisms. From our results, it can be noticed that our texture

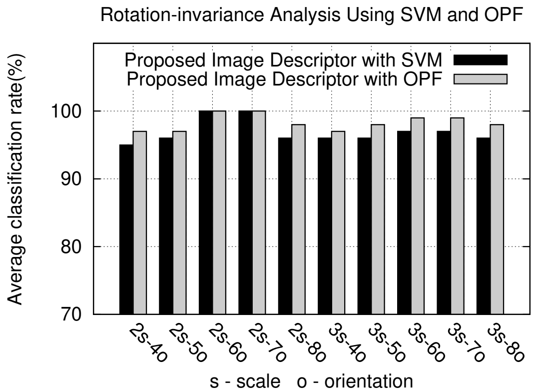


Figura 3.4: Recognition accuracy comparison of the OPF and SVM classifiers in *rotated-set* A using $(S = 2, 3)$ scales with $(K = 4, 5, 6, 7, 8)$ orientations for our rotation-invariant image descriptor.

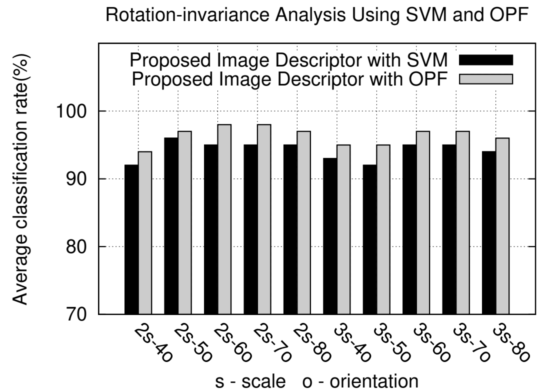


Figura 3.5: Recognition accuracy comparison of the OPF and SVM classifiers in *rotated-set* B using $(S = 2, 3)$ scales with $(K = 4, 5, 6, 7, 8)$ orientations for our rotation-invariant image descriptor.

image descriptor performs better regardless of the dataset used, or the image decomposition parameters considered during feature extraction (number of scales and orientations). Furthermore, the second series of experiments are summarized in Table 3.2. As it can be seen, the OPF classifier improves the recognition accuracies obtained by the SVM classifier in all of our experiments. By considering the summarized results, it can be shown that our proposed recognition system performs better than the previously mentioned approaches, which represent state-of-the-art methods.

Rotated Image Dataset	Feature vectors with different scales (s) and orientations (o)	Proposed Image Descriptor	Conv. Steerable Pyd. Dec.	Gabor Wavelets
A	(s=2;o=4,5,6,7,8)	**98.89%**	93.19%	93.19%
A	(s=3;o=4,5,6,7,8)	**98.61%**	92.36%	97.92%
B	(s=2;o=4,5,6,7,8)	**97.35%**	85.30%	91.29%
B	(s=3;o=4,5,6,7,8)	**96.74%**	85.76%	96.67%

Tabela 3.1: Mean recognition rates for the three different texture image descriptors using Gaussian-kernel Support Vector Machines as classifiers.

3.7 Conclusions

In this paper a new novel texture classification system was proposed. Its main features include: (1) a new rotation-invariant image descriptor, and (2) a novel multi-class recogni-

Rotated Image Dataset	Feature vectors with different scales (s) and orientations (o)	Proposed Image Descriptor using OPF	Proposed Image Descriptor using SVM
A	(s=2;o=4,5,6,7,8)	**98.89%**	95.89%
A	(s=3;o=4,5,6,7,8)	**98.61%**	97.99%
B	(s=2;o=4,5,6,7,8)	**97.35%**	92.30%
B	(s=3;o=4,5,6,7,8)	**96.74%**	96.70%

Tabela 3.2: Mean recognition rates for the proposed rotation-invariant texture image descriptor using both OPF and SVM classifiers.

tion method based on Optimum Path Forest. The proposed image descriptor exploits the discriminability properties of the Steerable Pyramid Decomposition for texture characterization. To obtain rotation-invariance, the dominant orientation of the input textures is found, so that feature elements are aligned according to this value. By doing this, a more reliable feature extraction process can be performed, since corresponding feature elements of distinct feature vectors, coincide with images at same orientations. In addition, our system adopted a novel approach for pattern classification based on Optimum Path Forest, which finds prototypes with none zero classification errors in the training sets and learns from errors in evaluation sets. By combining the discriminating power of our image descriptor and classifier, our system uses small size feature vectors to characterize texture images without compromising overall classification rates, being ideally for real-time applications.

Furthermore, we have demonstrated state-of-the-art results on two image datasets derived from the standard Brodatz database. For the first image dataset, our method obtained a mean classification rate of 98.89% in comparision with a mean accuracy of 93.19% obtained by both conventional Steerable Pyramid decomposition [124] and Gabor Wavelets [3]. For the second image dataset, our method achieved a mean classification of 97.35%, whereas the other two methods obtained respectively classification rates of 85.30% and 91.29%.

On the other side, the OPF multi-class classifier outperformed the SVM in the two datasets. It is a new promising graph tool for pattern recognition, which differs from traditional approaches, in that, it does not use the idea of feature space space geometry, therefore, better results in overlapped databases are achieved. Future work will include extending this method for scale-invariant texture recognition.

3.8 Acknowledgements

This work is partially supported by Microsoft Escience and Tablet PC Technology and Higher Education projects, CNPq (Proc. 134990/2005-6, 477039/2006-5 and 311309/2006-2), FAPESP, CAPES, and FAEPEX.

Capítulo 4

Learning How to Extract Rotation-Invariant and Scale-Invariant Features from Texture Images

4.1 Introduction

An important low-level image feature used in human perception as well as in recognition is texture. In fact, the study of texture has found several applications ranging from texture segmentation [120] to texture classification[y], synthesis [7, 147], and image retrieval [77, 125].

Although various authors have attempted to define what texture is [49,113], there still does not exist a commonly accepted definition. However, the basic property presented in every texture, consists in a small elementary pattern repeated periodically or quasi-periodically in a given region (pixel neighborhood) [54,58]. The repetition of those image patterns generate some visual cues, which can be identified, for example, as being directional or non-directional, smooth or rough, coarse or fine, uniform or non-uniform [145]. Figures 4.1- 4.4 show some examples of these types of visual cues. Note that each texture can be associated with one or more visual cues

Further, texture images are typically classified as being either natural or artificial. Natural textures are related to non-man-made objects and among others they include, for example, brick, grass, sand, and wood patterns. On the other side, artificial textures are related to man-made objects such as architectural, fabric, and metal patterns.

Regardless of its classification type, texture images may be characterized by their variations in scale or directionality. Scale variations imply that textures may look quite

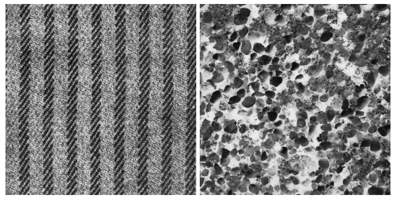


Figura 4.1: *directional vs. non-directional* visual cues.

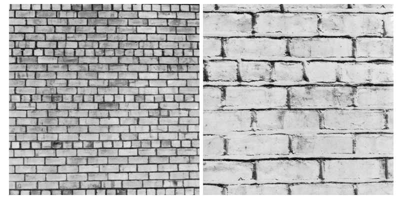


Figura 4.2: *smooth vs. rough* visual cues.

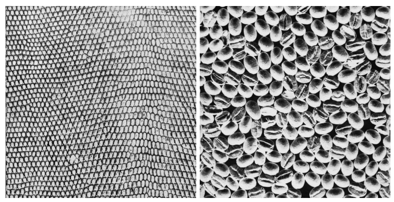


Figura 4.3: *fine vs. coarse* visual cues.

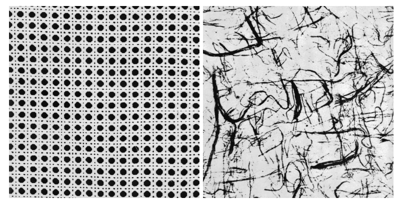


Figura 4.4: *uniform vs. non-uniform* visual cues.

different when varying the number of scales. This effect is analogous to increase or decrease the image resolution. The bigger or the smaller the scales are, the more different the images are. This characteristic is related to the *coarseness* presented in texture images and can be understood as the spatial repetition period of the local pattern [116]. Finer texture images are characterized by small repetition periods, whereas coarse textures present larger repetition periods. In addition, oriented textures may present different principal directions as the images rotate. This happens because textures are not always captured from the same viewpoint.

On the other hand, work on texture characterization can be divided into four major categories [41, 120]: structural, statistical, model-based and spectral. For structural methods, texture images can be thought as being a set of primitives with geometrical properties. Their objective is therefore to find the primitive elements as well as the formal rules of their spatial placements. Example of this kind of methods can be found in Julesz's [64] and Tüceryan's work [133]. In addition, statistical methods study the spatial gray level distribution in the textural patterns, so that statistical operations can be performed in the distributions of the local features computed at each pixel in the image. Statistical methods include among others the gray level co-occurrence matrix [47], second order spatial averages, and the auto-correlation function [18]. Further, the objective of the model-based methods is to capture the process that generated the texture patterns. Popular approaches in this category include Markov random fields [31, 38], fractal [111], and auto-regressive models [89]. Finally, spectral methods perform frequency analysis in the image signals to reveal specific features. Examples of this may include Law's [71] and Gabor filters [13, 55].

Although many of these techniques obtained good results, most of them have not been widely evaluated in *non-controlled* environments, which may be characterized by texture images having: (1) small inter-class variations, i.e, textures belonging to different classes may appear quite similar, especially in terms of their global patterns (coarseness, smoothness, etc.), and the patterns may present (2) image distortions such as rotations or scales. In this sense, texture pattern recognition is a still open task. The next challenge in texture classification should be, therefore, to achieve rotation-invariant and scale-invariant feature representations for *non-controlled* environments.

Some of these challenges are faced in this work. More specifically, we focus in feature representation and recognition. In feature representation, we wish to emphasize some open questions, such as: How to model the texture images so that the relevant information is captured despite of the image distortions?, and How to keep low dimensional feature vectors so that texture recognition applications are facilitated, where data storage capacity is a limitation? In feature recognition, we wish to choose a technique that handles multiple non-separable classes with minimal computational time and supervision. To deal with the challenges in feature extraction, we propose a new texture image descriptor based on Steerable Pyramid Decomposition, which encodes the relevant texture information in small-size feature vectors including rotation-invariant and scale-invariant characterizations. To address the feature recognition requirements, we are using a novel multi-class object recognition method based on the Optimum-Path Forest [109].

Roughly speaking, a Steerable Pyramid is a method by which images are decomposed into a set of multi-scale, and multi-orientation image subbands, where the basis functions are directional derivative operators [37]. Our motivation in using Steerable Pyramids relies on that, unlike other image decomposition methods, the feature coefficients are less affected by image distortions. Furthermore, the Optimum-Path Forest Classifier is a recent approach that handles non-separable classes, without the necessity of using boosting procedures to increase its performance, resulting thus in a faster and more accurate classifier for object recognition. By combining the discriminating power of our image descriptor and classifier, our system uses small-size feature vectors to characterize texture images without compromising overall classification rates. In this way, texture classification applications, where data storage capacity is a limitation, are further facilitated.

A previous version of our texture descriptor has been proposed for Content-Based Image Retrieval [92] and Texture Recognition [96], but using only rotation-invariant properties. In the present work, the proposed descriptor has not only rotation-invariant properties, but also scale-invariant properties. The Optimum-Path Forest classifier was first presented in [107] and first evaluated for texture recognition in [96]. Improvements in its learning algorithm and evaluation with several datasets have been made in [109], for other properties rather than texture. The present work is using this most recent version of the Optimum-Path Forest classifier for texture recognition. We are providing more details about the methods, experiments with more datasets, and a more in deep analysis of the results: rotation- and scale-invariance analysis, accuracy of classification with different descriptors and classifiers, and the mean computational time of the proposed system.

The outline of this work is as follows. In the next section, we briefly review the fundamentals of the Steerable Pyramid Decomposition. Section 2.3 describes how texture images are characterized to obtain rotation-invariant, and scale-invariant representations. Section 3.4 introduces the Optimum Path Forest classifier method. The experimental setup conducted in our study is presented in Section 4.5. In section 4.6, experimental results on several datasets are given and are used to demonstrate the recognition accuracy of our system. Comparisons with state-of-the-art texture feature representations and classifiers are further discussed. Finally, some conclusions are drawn in Section 4.7.

4.2 Steerable Pyramid Decomposition

The Steerable Pyramid Decomposition is a linear multi-resolution image decomposition method by which an image is subdivided into a collection of subbands localized at different scales and orientations [37]. Using a high- and low-pass filter (H_0, L_0) the input image is initially decomposed into two subbands: a high- and a low-pass subband, respectively. Further, the low-pass subband is decomposed into K-oriented band-pass portions $B_0, \ldots, B_{K-1}$, and into a lowpass subband L_1. The decomposition is done recursively by subsampling the lower low-pass subband (L_S) by a factor of 2 along the rows and columns. Each recursive step captures different directional information at a given scale. Considering the polar-separability of the filters in the Fourier domain, the first low- and high-pass filters, are defined as [115]:

$$\begin{aligned} L_0(r,\theta) = & \quad L\left(\tfrac{r}{2},\theta\right)/2 \\ H_0(r,\theta) = & \quad H\left(\tfrac{r}{2},\theta\right) \end{aligned} \tag{4.1}$$

where r,θ are the polar frequency coordinates. The raised cosine low-, and high-pass transfer functions denoted as L, H, respectively, are computed as follows:

$$L(r,\theta) = \begin{cases} 2 & r \leq \frac{\pi}{4} \\ 2\cos\left(\frac{\pi}{2}\log_2\left(\frac{4r}{\pi}\right)\right) & \frac{\pi}{4} < r < \frac{\pi}{2} \\ 0 & r \geq \frac{\pi}{2} \end{cases} \tag{4.2}$$

$$B_k(r,\theta) = H(r)G_k(\theta), \qquad k \in [0, K-1] \tag{4.3}$$

$B_k(r,\theta)$ represents the *k-th* directional bandpass filter used in the iterative stages, with radial and angular parameters, defined as:

$$H(r,\theta) = \begin{cases} 1 & r \geq \frac{\pi}{4} \\ \cos\left(\frac{\pi}{2}\log_2\left(\frac{2r}{\pi}\right)\right) & \frac{\pi}{4} < r < \frac{\pi}{2} \\ 0 & r \leq \frac{\pi}{2} \end{cases} \quad (4.4)$$

$$G_k(\theta) = \begin{cases} \alpha_K\left(\cos\left(\theta - \frac{\pi k}{K}\right)\right)^{K-1} & \left|\theta - \frac{\pi k}{K}\right| < \frac{\pi}{2} \\ 0 & \text{otherwise} \end{cases} \quad (4.5)$$

where $\alpha_k = 2^{(k-1)}\frac{(K-1)!}{\sqrt{K[2(K-1)]!}}$.
Figure 4.5 depicts a Steerable Pyramid Decomposition using only one scale and n orientations.

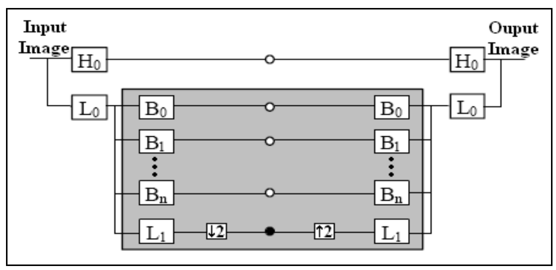


Figura 4.5: First level Steerable Pyramid Decomposition using n oriented band-pass filters.

4.3 Texture feature representation

This section describes the proposed modification of Steerable Pyramid Decomposition to obtain rotation-invariant, and scale-invariant representations, used further to characterize the texture images.

4.3.1 Texture representation

Roughly speaking, texture images can be seen as a set of basic repetitive primitives characterized by their spatial homogeneity [28]. By applying statistical measures, this information is extracted, and used to capture the relevant image content into feature vectors. More precisely, by considering the presence of homogeneous regions in texture images, we use the mean (μ_{mn}) and standard deviation (σ_{mn}) of the energy distribution of the filtered images (S_{mn}). Given an image $I(x,y)$, its Steerable Pyramid Decomposition is defined as:

$$S_{mn}(x,y) = \sum_{x_1}\sum_{y_1} I(x_1,y_1)B_{mn}(x-x_1,y-y_1) \tag{4.6}$$

where B_{mn} denotes the directional bandpass filters at stage $m = 0,1,\ldots,S-1$, and orientation $n = 0,1,\ldots,K-1$. The energy distribution $(E(m,n))$ of the filtered images at scale m, and at orientation n is defined as:

$$E(m,n) = \sum_{x}\sum_{y} |S_{mn}(x,y)| \tag{4.7}$$

Additionally, the mean (μ_{mn}) and standard deviation (σ_{mn}) of the energy distributions are found as follows:

$$\mu_{mn} = \frac{1}{MN}E_{mn}(x,y) \tag{4.8}$$

$$\sigma_{mn} = \sqrt{\frac{1}{MN}\sum_{x}\sum_{y}\left(|S_{mn}(x,y)| - \mu_{mn}\right)^2} \tag{4.9}$$

where M and N denote the height and width of the input image, respectively. The corresponding feature vector $(\vec{f})$ is defined by using the mean and standard deviation as feature elements. It is denoted as:

$$\vec{f} = [\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \ldots, \mu_{S-1K-1}, \sigma_{S-1K-1}] \tag{4.10}$$

The dimensionality of the feature vectors depends on the number of scales (S) and on the number of orientations (O) considered during image decomposition. The feature vector dimensionality is computed by multiplying the number of scales and orientations by factor of 2 $(2 \times O \times S)$. This factor corresponds to the mean and standard deviation computed in each filtered image.

4.3.2 Rotation-invariant representation

Rotation-invariant representation is achieved by computing the dominant orientation of the texture images followed by feature alignment. The **dominant orientation** (DO) is defined as the orientation with the highest total energy across the different scales considered during image decomposition [3]. It is computed by finding the highest accumulated energy for the K different orientations considered during image decomposition:

$$DO_i = max\left\{E_0^{(R)}, E_1^{(R)}, \ldots, E_{K-1}^{(R)}\right\} \tag{4.11}$$

where i is the index where the dominant orientation appeared, and:

$$E_n^{(R)} = \sum_{m=0}^{S-1} E(m,n), \quad n = 0,1,\ldots,K-1. \tag{4.12}$$

Note that each $E_n^{(R)}$ covers a set of filtered images at different scales but at same orientation.

Finally, rotation-invariance is obtained by shifting circularly feature elements within the same scales, so that first elements at each scale correspond to dominant orientations. As an example, let $\vec{f}$ be a feature vector obtained by using a Pyramid Decomposition with $S = 2$ scales, and $K = 3$ orientations:

$$\begin{aligned}\vec{f} = \; & [\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \mu_{02}, \sigma_{02}; \\ & \mu_{10}, \sigma_{10}, \mu_{11}, \sigma_{11}, \mu_{12}, \sigma_{12}]\end{aligned} \tag{4.13}$$

Now suppose that the dominant orientation appears at index $i = 1$ ($DO_{i=1}$), thus the rotation-invariant feature vector, after feature aligment, is represented as follows:

$$\begin{aligned}\vec{f}^R = \; & [\mu_{01}, \sigma_{01}, \mu_{02}, \sigma_{02}, \mu_{00}, \sigma_{00}; \\ & \mu_{11}, \sigma_{11}, \mu_{12}, \sigma_{12}, \mu_{10}, \sigma_{10}]\end{aligned} \tag{4.14}$$

4.3.3 Scale-invariant representation

Similarly, scale-invariant representation is achieved by finding the scale with the highest total energy across the different orientations (**dominant scale**). For this purpose,the dominant scale (DS) at index i is computed as follows:

$$DS_i = max\left\{E_0^{(S)}, E_1^{(S)}, \ldots, E_{S-1}^{(S)}\right\} \tag{4.15}$$

where $E_m^{(S)}$ denotes the accumulated energies across the S different scales:

$$E_m^{(S)} = \sum_{n=0}^{K-1} E(m, n), \quad m = 0, 1, \ldots, S-1. \tag{4.16}$$

Note that each $E_m^{(S)}$ covers a set of filtered images at different orientations for each scale. As an example, let $\vec{f}$ be, again, the feature vector obtained by using a Pyramid Decomposition with $S = 2$ scales, and $K = 3$ orientations:

$$\begin{aligned}\vec{f} = \; & [\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \mu_{02}, \sigma_{02}; \\ & \mu_{10}, \sigma_{10}, \mu_{11}, \sigma_{11}, \mu_{12}, \sigma_{12}]\end{aligned} \tag{4.17}$$

By supposing that the dominant scale was found at index $i = 1$ (second scale in the image decomposition), its scale-invariant version, after feature alignment, is defined as:

$$\begin{aligned}\vec{f}^S = \; & [\mu_{10}, \sigma_{10}, \mu_{11}, \sigma_{11}, \mu_{12}, \sigma_{12}; \\ & \mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \mu_{02}, \sigma_{02}]\end{aligned} \tag{4.18}$$

For both rotation-invariant, and scale-invariant representations, the feature alignment process is based on the assumption that to classify textures, images should be rotated so that their dominant orientations/scales are the same. Further, it has been proved that image rotation in spatial domain is equivalent to circular shift of feature vector elements [149].

4.4 Texture feature recognition

This section aims to describe the most recent version of the Optimum-Path Forest (OPF) classifier [109], which is an important part of the texture recognition system proposed in this work. Previous works have demonstrated that OPF can be more effective and much faster than Artificial Neural Networks [108] and Support Vector Machines [107–109]. The OPF approach works by modeling the patterns as being nodes of a graph in the feature space, where every pair of nodes is connected by an arc (complete graph). This classifier creates a discrete optimal partition of the feature space such that any unknown sample can be classified according to this partition. This partition is an optimum-path forest computed in $\Re^n$ by the image foresting transform (IFT) algorithm [33]. The OPF classifier extends the IFT from the image domain to the feature space, where the samples may be images, contours, or any other abstract entities.

Let Z_1, Z_2, and Z_3 be respectively the training, evaluation, and test sets with $|Z_1|$, $|Z_2|$, and $|Z_3|$ samples, respectively. Let $\lambda(s)$ be the function that assigns the correct label i, $i = 1, 2, \ldots, c$, from class i to any sample $s \in Z_1 \cup Z_2 \cup Z_3$. Z_1 and Z_2 are labeled sets used to the design of the classifier and the unseen set Z_3 is used to compute the final accuracy of the classifier. Let $S \subset Z_1$ be a set of prototypes of all classes (i.e., key samples that best represent the classes). Let v be an algorithm which extracts n attributes (texture properties) from any sample $s \in Z_1 \cup Z_2 \cup Z_3$ and returns a vector $\vec{v}(s) \in \Re^n$. The distance $d(s, t)$ between two samples, s and t, is the one between their feature vectors $\vec{v}(s)$ and $\vec{v}(t)$ (e.g., Euclidean or any other valid metric).

Let (Z_1, A) be a complete graph whose the nodes are the samples in Z_1. We define a path as being a sequence of distinct samples $\pi = \langle s_1, s_2, \ldots, s_k \rangle$, where $(s_i, s_{i+1}) \in A$ for $1 \leq i \leq k-1$. A path is said *trivial* if $\pi = \langle s_1 \rangle$. We assign to each path π a cost $f(\pi)$ given by a path-cost function f. A path π is said optimum if $f(\pi) \leq f(\pi')$ for any other path π', where π and π' end at a same sample s_k. We also denote by $\pi \cdot \langle s, t \rangle$ the concatenation of a path π with terminus at s and an arc (s, t). The OPF algorithm uses the path-cost function f_{max}, for the reason explained in Section 3.4.1.

$$
\begin{aligned}
f_{max}(\langle s \rangle) &= \begin{cases} 0 & \text{if } s \in S, \\ +\infty & \text{otherwise} \end{cases} \\
f_{max}(\pi \cdot \langle s, t \rangle) &= \max\{f_{max}(\pi), d(s,t)\}
\end{aligned} \tag{4.19}
$$

We can observe that $f_{max}(\pi)$ computes the maximum distance between adjacent samples in π, when π is not a trivial path.

The OPF algorithm assigns one optimum path $P^*(s)$ from S to every sample $s \in Z_1$, forming an optimum-path forest P (a function with no cycles which assigns to each $s \in Z_1 \backslash S$, its predecessor $P(s)$ in $P^*(s)$, or a marker *nil* when $s \in S$). Let $R(s) \in S$ be the root of $P^*(s)$ which can be reached from $P(s)$. The OPF algorithm computes for each $s \in Z_1$, the cost $C(s)$ of $P^*(s)$, the label $L(s) = \lambda(R(s))$, and the predecessor $P(s)$, as follows.

Algorithm 2. – OPF ALGORITHM

INPUT:	A λ-labeled training set Z_1, prototypes $S \subset Z_1$ and the pair (v, d) for feature vector and distance computations.
OUTPUT:	Optimum-path forest P, cost map C and label map L.
AUXILIARY:	Priority queue Q and cost variable cst.

1. *For each* $s \in Z_1 \backslash S$, *set* $C(s) \leftarrow +\infty$.
2. *For each* $s \in S$, *do*
3. $C(s) \leftarrow 0$, $P(s) \leftarrow nil$, $L(s) \leftarrow \lambda(s)$, *and insert* s *in* Q.
4. *While* Q *is not empty, do*
5.   *Remove from* Q *a sample* s *such that* $C(s)$ *is minimum.*
6.   *For each* $t \in Z_1$ *such that* $t \neq s$ *and* $C(t) > C(s)$, *do*
7.   *Compute* $cst \leftarrow \max\{C(s), d(s,t)\}$.
8.   *If* $cst < C(t)$, *then*
9.   *If* $C(t) \neq +\infty$, *then remove* t *from* Q.
10. $P(t) \leftarrow s$, $L(t) \leftarrow L(s)$, $C(t) \leftarrow cst$,
11.   *and insert* t *in* Q.

Lines $1-3$ initialize maps and insert prototypes in Q. The main loop computes an optimum path from S to every sample s in a non-decreasing order of cost (Lines $4-11$). At each iteration, a path of minimum cost $C(s)$ is obtained in P when we remove its last node s from Q (Line 5). Lines $8-11$ evaluate if the path that reaches an adjacent node t through s is cheaper than the current path with terminus t and update the position of t in Q, $C(t)$, $L(t)$ and $P(t)$ accordingly. The label $L(s)$ may be different from $\lambda(s)$, leading to classification errors in Z_1. The training finds prototypes with zero classification errors in Z_1, as follows.

4.4.1 Training phase

We say that S^* is an optimum set of prototypes when Algorithm 1 propagates the labels $L(s) = \lambda(s)$ for every $s \in Z_1$. S^* can be found by exploiting the theoretical relation between minimum-spanning tree (MST) and optimum-path tree for f_{max} [1]. The training essentially consists of finding S^* and an OPF classifier rooted at S^*.

By computing an MST in the complete graph (Z_1, A), we obtain a connected acyclic graph whose nodes are all samples in Z_1 and the arcs are undirected and weighted by the distance d between the adjacent sample feature vectors. This spanning tree is optimum in the sense that the sum of its arc weights is minimum as compared to any other spanning tree in the complete graph. In the MST, every pair of samples is connected by a single path which is optimum according to f_{max}. The optimum prototypes are the closest elements in the MST with different labels in Z_1. By removing the arcs between different classes, their adjacent samples become prototypes in S^* and Algorithm 1 can compute an optimum-path forest with zero classification errors in Z_1.

It is not difficult to see that the optimum paths between classes should pass through the same removed arcs of the minimum-spanning tree. The choice of prototypes as described above blocks these passages, avoiding samples of any given class be reached by optimum paths from prototypes of other classes. Given that several methods for graph-based clustering are based on MST, the relation between MST and minimum-cost path tree for $f_{\max}$ [1] makes interesting connections among the supervised OPF classifier, these unsupervised approaches and the previous works on watershed-based/fuzzy-connected segmentations [4, 5, 9, 33, 79, 122, 138].

4.4.2 Classification

For any sample $t \in Z_3$, the OPF consider all arcs connecting t with samples $s \in Z_1$, as though t were part of the graph. Considering all possible paths from S^* to t, we wish to find the optimum path $P^*(t)$ from S^* and label t with the class $\lambda(R(t))$ of its most strongly connected prototype $R(t) \in S^*$. This path can be identified incrementally, by evaluating the optimum cost $C(t)$ as

$$C(t) = \min\{\max\{C(s), d(s,t)\}\}, \ \forall s \in Z_1. \tag{4.20}$$

Let the node $s^* \in Z_1$ be the one that satisfies the above equation (i.e., the predecessor $P(t)$ in the optimum path $P^*(t)$). Given that $L(s^*) = \lambda(R(t))$, the classification simply assigns $L(s^*)$ to t. An error occurs when $L(s^*) \neq \lambda(t)$.

4.4.3 Learning algorithm

The performance of the OPF classifier improves when the closest samples from different classes are included in Z_1, because the method finds prototypes that will work as sentinels in the frontier between classes.

We propose a simple but efficient algorithm that identifies the best prototypes in a few number of iterations. We suppose that these prototypes are the misclassified elements in Z_2. So, we randomly replace them by not prototypes samples from Z_1. The algorithm outputs a *learning curve* over T iterations, which reports the accuracy values of each instance of classifier during learning, and the final OPF classifier. Algorithm 3 describes this procedure.

Algorithm 3. – LEARNING ALGORITHM

INPUT:	Training and evaluation sets labeled by λ, Z_1 and Z_2, number T of iterations, and the pair (v, d) for feature vector and distance computations.
OUTPUT:	Learning curve $\mathcal{L}$ and the last OPF classifier, represented by the predecessor map P, cost map C, and label map L.
AUXILIARY:	False positive and false negative arrays, FP and FN, of sizes c, and list LM of misclassified samples.

1. *For each iteration* $I = 1, 2, \ldots, T$, *do*
2. $\quad LM \leftarrow \emptyset$
3. $\quad$ *Compute* $S^* \subset Z_1$ *as in Section 4.4.1 and* P, L, C
4. $\quad$ *by Algorithm 2.*
5. $\quad$ *For each class* $i = 1, 2, \ldots, c$, *do*
6. $\quad\quad FP(i) \leftarrow 0$ *and* $FN(i) \leftarrow 0$.
7. $\quad$ *For each sample* $t \in Z_2$, *do*
8. $\quad\quad$ *Find* $s^* \in Z_1$ *that satisfies Equation 4.20.*
9. $\quad\quad$ *If* $L(s^*) \neq \lambda(t)$, *then*
10. $\quad\quad\quad FP(L(s^*)) \leftarrow FP(L(s^*)) + 1$.
11. $\quad\quad\quad FN(\lambda(t)) \leftarrow FN(\lambda(t)) + 1$.
12. $\quad\quad\quad LM \leftarrow LM \cup t$.
13. $\quad$ *Compute* $\mathcal{L}(I)$ *by Equation 4.23.*
14. $\quad$ *While* $LM \neq \emptyset$
15. $\quad\quad LM \leftarrow LM \backslash t$
16. $\quad\quad$ *Replace* t *by randomly objects of the same class*
17. $\quad\quad$ *in* Z_1, *except the prototypes.*

Lines $3 - 4$ initializes the false positive and false negative vectors. The classification of each sample is performed by Lines $6 - 11$, updating the false positive and false negative vectors. Misclassified samples are stored in vector LM in Line 11, and the inner loop in Lines $13-16$ changes the misclassified samples in Z_2 by non prototypes samples in Z_1.

Line 11 computes the accuracy at iteration I and stores it in the learning curve $\mathcal{L}$. The accuracy $\mathcal{L}(I)$ of a given iteration I, $I = 1, 2, \ldots, T$, is measured by taking into account that the classes may have different sizes in Z_2 (similar definition is applied for Z_3). Let $NZ_2(i)$, $i = 1, 2, \ldots, c$, be the number of samples in Z_2 from each class i. We define

$$e_{i,1} = \frac{FP(i)}{|Z_2| - |NZ_2(i)|} \quad \text{and} \quad e_{i,2} = \frac{FN(i)}{|NZ_2(i)|}, \; i = 1, \ldots, c \tag{4.21}$$

where $FP(i)$ and $FN(i)$ are the false positives and false negatives, respectively. That is, $FP(i)$ is the number of samples from other classes that were classified as being from the class i in Z_2, and $FN(i)$ is the number of samples from the class i that were incorrectly classified as being from other classes in Z_2. The errors $e_{i,1}$ and $e_{i,2}$ are used to define

$$E(i) = e_{i,1} + e_{i,2}, \tag{4.22}$$

where $E(i)$ is the partial sum error of class i. Finally, the accuracy $\mathcal{L}(I)$ of the classification is written as

$$\mathcal{L}(I) = \frac{2c - \sum_{i=1}^{c} E(i)}{2c} = 1 - \frac{\sum_{i=1}^{c} E(i)}{2c}. \tag{4.23}$$

4.5 Experimental setup

4.5.1 Datasets

To evaluate the accuracy of our system, thirteen texture images obtained from the standard Brodatz database were selected. Before being digitized, each of the 512×512 texture images was rotated at different degrees [103]. Figure 4.6 displays the non-rotated version of each of the texture images.

From this database, three different image datasets were generated: *non-distorted*, *rotated-set*, and *scaled-set*. The non-distorted image dataset was constructed from texture patterns at 0 degrees. Each texture image was partitioned into sixteen 128×128 non-overlapping subimages. Thus, this dataset comprises 208 (13×16) different images. Images belonging to this dataset will be used in the learning stage of our classifier. Note that in previous works related to texture recognition [44, 130], rotated or scaled-versions of the patterns were included in both the training and classification phases [69]. However, more recently works suggest that the recognition algorithms should perform well, even by having during the training phase non-distorted training samples, this means, patterns without rotations or scales [104, 114].

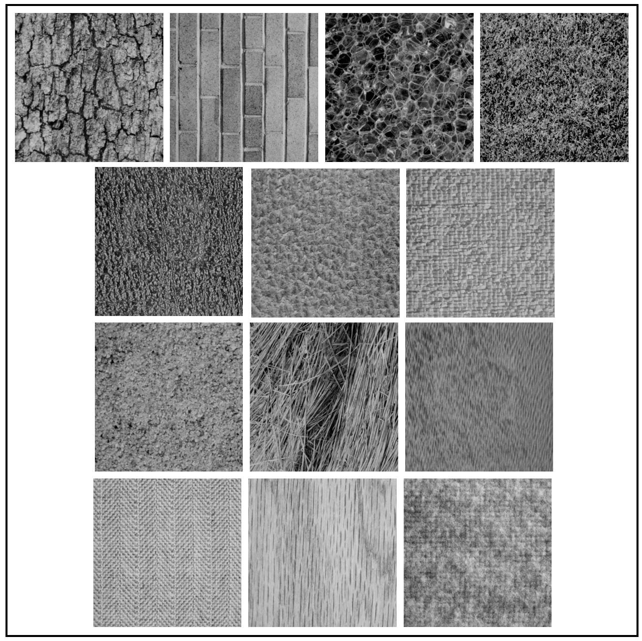


Figura 4.6: Texture images from the Brodatz dataset used in our experiments. From left to right, and from top to bottom, they include: Bark, Brick, Bubbles, Grass, Leather, Pigskin, Raffia, Sand, Straw, Water, Weave, Wood, and Wool.

The second image dataset referred to as *rotated-set* was generated to evaluate the rotation-invariance capabilities of our approach. It is further subdivided into two datasets: *rotated-set A* and *rotated-set B*. The *rotated image dataset A* was generated by selecting the four 128×128 innermost subimages from texture images at $0, 30, 60$, and 120 degrees. A total number of 208 images was generated ($13 \times 4 \times 4$). In addition, in the case of the *rotated image dataset B*, we selected the four 128×128 innermost subimages of the rotated image textures (512×512) at $0, 30, 60, 90, 120, 150$ and 200 degrees. This led to 364 ($13 \times 4 \times 4$) dataset images. The first dataset was initially used to test our system under the presence of few texture oriented images, whereas the second one was used to show how our systems performs by increasing the number of texture oriented images.

On the other side, the *scaled image dataset* was partitioned into two datasets: *scaled-set A* and *scaled-set B*. In the *scaled-set A*, the 512×512 non-rotated textures were first partitioned into four 256×256 non-overlapping subimages. Each partitioned subimage was further scaled by using four different factors, ranging from 0.6 to 0.9 with 0.1 interval. This led to 208 ($13 \times 4 \times 4$) scaled images. To generate the *scaled-set B*, each of the four partitioned subimages was scaled by using seven different factors, ranging from 0.6 to 1.2 with 0.1 interval. In this way, 364 ($13 \times 4 \times 7$) scaled images were generated.

4.5.2 Similarity Measure for Classification

Similarity between images is obtained by computing the distance of their corresponding feature vectors (Recall Section 2.3). The smaller the distance, the more similar the images. Given the query image (i), and the target image (j) in the dataset, the distance between the two patterns is defined as [87]:

$$d(i,j) = \sum_m \sum_n d_{mn}(i,j) \tag{4.24}$$

where:

$$d_{mn}(i,j) = \left| \frac{\mu^i_{mn} - \mu^j_{mn}}{\alpha(\mu_{mn})} \right| + \left| \frac{\sigma^i_{mn} - \sigma^j_{mn}}{\alpha(\sigma_{mn})} \right|, \tag{4.25}$$

$\alpha(\mu_{mn})$ and $\alpha(\sigma_{mn})$ denote the standard deviations of the respective features over the entire dataset. They are used for feature normalization purposes.

4.6 Experimental Results

Three series of experiments were conducted to demonstrate the discriminating power of our system for recognizing texture patterns. By considering that a recognition system is comprised of two mainly parts (feature extraction module as well as feature recognizer module), each of those parts were evaluated.

In the first series of experiments (Subsection 3.6.1), we first evaluated the effectiveness of the proposed rotation-invariant feature representation against two other approaches: the conventional Pyramid Decomposition [124] and with a recent proposal based on Gabor Wavelets [3]. To evaluate the effectiveness of the feature recognizer module, we compared the recognition accuracy of the novel OPF multi-class classifier against the well known Support Vector Machines technique. For those purposes, we used the *rotated image datasets A* and *B*.

The second series of experiments (Subsection 4.6.2) are used to evaluate the scale-invariant properties in our feature extraction module. Effectiveness of the multiclass recognition method under the presence of scale-invariant features are further discussed. Again, we used the conventional Steerable Pyramid Decomposition [124] and the Gabor Wavelets [87] as references for comparing the scale-invariant properties of our method. SVMs are used for evaluating the classification accuracy of our feature recognizer module. Further, *scaled image datasets A* and *B* were used in this set of experiments.

In both series of experiments, we used Steerable Pyramids having different decomposition levels ($S = 2, 3, 4$) at several orientations ($K = 4, 5, 6, 7, 8$). Our experiments agree with [30] in that the most relevant textural information in images is contained in the first two levels of decomposition, since little recognition improvement is achieved by varying the number of scales during image decomposition. Therefore, we focus our discussions on image decompositions having ($S = 2, 3$) scales.

Given that the performance of the OPF classifier can increase using a third set in a learning algorithm (Subsection 3.4.3), we also employed this same procedure to the SVM approach. The constraints in Lines $16-17$ of the Algorithm 3 refer to keep the prototypes out of the sample interchanging process between Z_1 and Z_2 for the OPF. We do the same with the support vectors in SVM. However, they may be selected for interchanging in future iterations if they are no longer prototypes or support vectors. For SVM, we use the LibSVM package [19] with Radial Basis Function (RBF) kernel, parameter optimization and the one-versus-one strategy for the multi-class problem to implement Line 3.

The experiments evaluate the accuracy on Z_3 and the computational time of each classifier, OPF and SVM. In all experiments, the datasets were divided into three parts: a training set Z_1 with 20% of the samples, an evaluation set Z_2 with 30% of the samples, and a test set Z_3 with 50% of the samples. These samples were ramdomly selected and each experiment was repeated 10 times with different sets Z_1, Z_2 and Z_3 to compute the mean accuracy.

Recall that an important motivation in our study is to use small size feature vectors, in order: (1) to show that the recognition accuracy of our approach is not compromised, and (2) to facilitate texture recognition applications where data storage capacity is a limitation.

4.6.1 Effectiveness of the rotation invariance representation

To analyze the texture characterization capabilities of our feature extraction method against the conventional Pyramid Decomposition and the Gabor Wavelets, we used Gaussian kernel Support Vector Machines (SVMs) as texture classification mechanisms.[1].

Figure 4.7 compares the recognition accuracy obtained by those three methods in the *rotated dataset A*, whereas Figure 4.8 depicts the recognition accuracy obtained in the *rotated dataset B*. From both Figures, it can be seen that our image descriptor outperforms mostly the other two approaches, regardless of the number of scales or orientations considered during feature vector extraction.

In the case of the *rotated dataset A*, the higher classification accuracies achieved by our method were obtained by using 7 orientations, which corresponds to image rotations in steps of 25.71°. By considering two and three decomposition levels ($s = 2, 3$), those accuracies are respectively 100% and 97.31%. The equivalent classification accuracies obtained by the Gabor Wavelets are: 90.36% and 93.90% ($s = 2, 3; o = 7$), whereas for

[1]Note that the SVM parameters were optimized by using the cross-validation method.

the conventional Steerable Pyramid those accuracies are: 89.67% and 90.36%. Note that, the classification accuracies obtained by using $o = 6, 7, 8$ orientations are very close to each other. Therefore, to guarantee low dimensionality feature vectors, we set $s = 2$ and $o = 6$ as the most appropriate parameter combinations for our rotation-invariant image descriptor.

In the case of the *rotated-set B*, the higher classification accuracies achieved by our descriptor were again obtained by using 7 orientations. Classification rates of 95.86% and 95.73% correspond respectively to feature vectors with $s = 2, 3$ scales and $o = 7$ orientations. Further, it is found that both Gabor Wavelets and Conventional Steerable Pyramid Decomposition present lower classification rates, being respectively 91.05% and 95.35% for the first method, and 84.22%, 84.23% for the second one. As stated in the results obtained in *rotated dataset A*, the classification accuracies are very close to each other, when using $o = 6, 7$ or $o = 8$ orientations. From those results, we can reinforce that the most appropriate parameter settings for our descriptor are $s = 2$ scales and $o = 6$ orientations.

Furthermore, from the bar graphs shown in Figures 4.7 and 4.8, the highest classification rate obtained by the Gabor method is as good as the one obtained by our descriptor. However, this rate is obtained at $s = 3$ scales, whereas our proposed descriptor achieves the same performance using only $s = 2$ scales. In this sense, an important advantage of our method is its high performance rate at low size feature vectors.

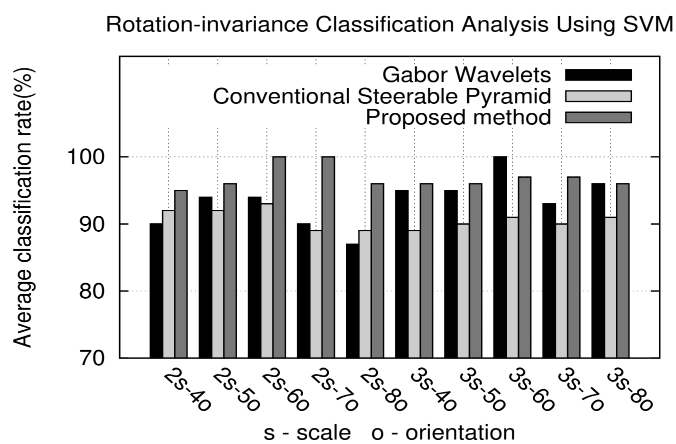


Figura 4.7: Classification accuracy comparison using the SVM classifier obtained in **rotated dataset A** using $(S = 2, 3)$ scales with $(K = 4, 5, 6, 7, 8)$ orientations for Gabor Wavelets, conventional Steerable Pyramid decomposition and our method.

Our objective now is to demonstrate the recognition improvement of our novel classifier over the SVM approach. From Figure 4.9 it can be seen, that for almost all feature extraction configurations, the recognition rates of the OPF classifier are higher than those

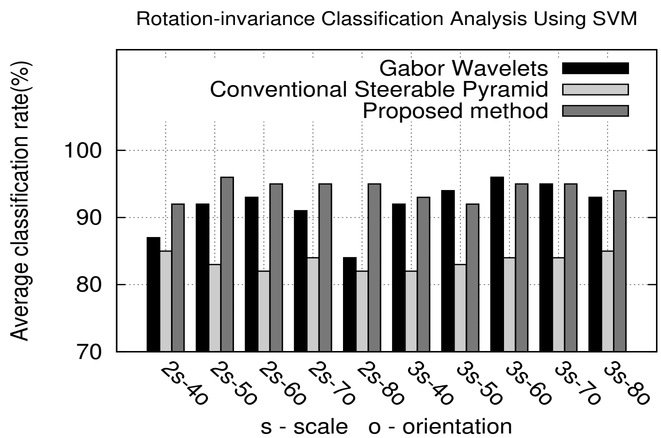


Figura 4.8: Classification accuracy comparison using the SVM classifier obtained in **rotated dataset B** using ($S = 2, 3$) scales with ($K = 4, 5, 6, 7, 8$) orientations for Gabor Wavelets, conventional Steerable Pyramid decomposition and our method.

of the SVM classifier. The latter method presents the same recognition rates as the ones of the OPF classifier when using $s = 2$ scales and $o = 6, 7$ orientations. In the case of the image *rotated dataset B*, our classifier yields better recognition rates for all feature extraction configurations (See Figure 4.10). By considering that it was found that the most appropriate parameter settings for our descriptor are $s = 2$ scales and $o = 6$ orientations, it is worth to mention that, by using this configuration, the recognition accuracy obtained by the OPF classifier is 98.49% in comparison with the corresponding accuracy of 95.48% obtained by the SVM classifier.

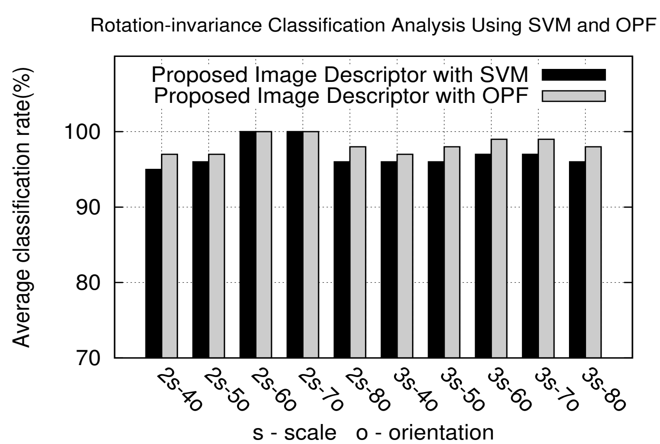


Figura 4.9: Classification accuracy comparison using the SVM classifier obtained in **rotated dataset A** using ($S = 2, 3$) scales with ($K = 4, 5, 6, 7, 8$) orientations for Gabor Wavelets, conventional Steerable Pyramid decomposition and our method.

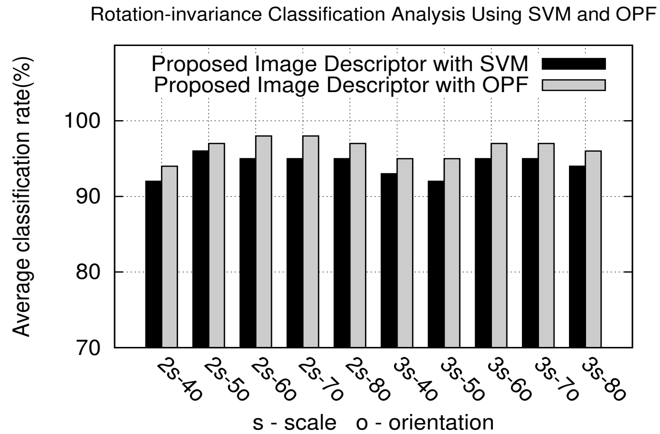


Figura 4.10: Classification accuracy comparison using the SVM classifier obtained in **rotated dataset B** using $(S = 2, 3)$ scales with $(K = 4, 5, 6, 7, 8)$ orientations for Gabor Wavelets, conventional Steerable Pyramid decomposition and our method.

4.6.2 Effectiveness of the scale invariance representation

Figures 4.11 and 4.12 display the classification accuracy of our scale-invariant image descriptor against the conventional Pyramid Decomposition and the Gabor Wavelets in the *scaled image datasets* A and B, respectively. Those Figures demonstrate the classification accuracy improvement of our image descriptor over both methods.

From Figure 4.11 it can be noticed that by using just $s = 2$ scales and $o = 7$ orientations, our feature extraction algorithm achieves a classification rate of 100%. This same rate is achieved by the other two methods, but at the cost of having larger image feature vectors. To obtain a classification rate of 100% both Pyramid Decomposition and Gabor Wavelets need at least $s = 3$ scales. Recalling Subsection 4.3.1, the feature vector dimensionality is obtained by multiplying the number of scales and orientations by a factor of 2, since we considered the mean and standard deviation as feature components. In this way, the dimensionality of our feature vectors is of size of $28(2 \times 2 \times 7)$ elements, in comparison with a size of $42(2 \times 3 \times 7)$ elements of their analogous feature vectors. By considering that the typical storage space of a float number is equal to 8 bytes, each of our feature vectors requires only 224 bytes to be stored, in comparison with the 336 bytes required for their analougus feature vectors. In this way, our image descriptor requires 66.7% less storage space than the one belonging to the compared descriptors.

By analyzing the classification accuracies depicted in Figure 4.12, we can notice again that our image features perform better than the other two methods. However, the main differences between the results presented in Figures 4.11 and 4.12 is that in the case of the *scale dataset B* all methods achieved higher classification rates. The reason for this lies in the tested texture dataset, which has more discriminative samples to be used during the training phase of the classifier. This can be thought as having sufficient discriminatory training data regardless of the testing data size.

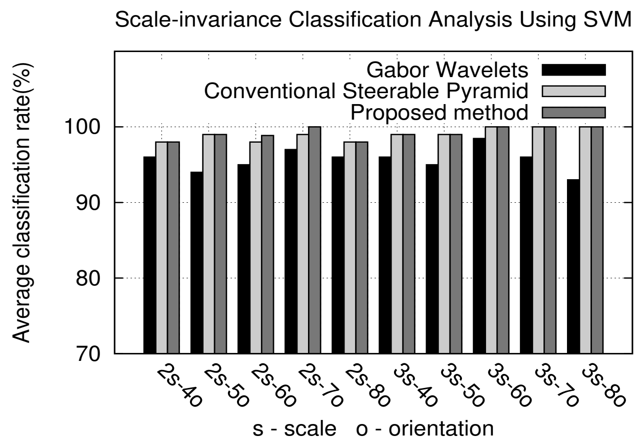


Figura 4.11: Classification accuracy comparison using the SVM classifier obtained in **scaled dataset A** using $(S = 2, 3)$ scales with $(K = 4, 5, 6, 7, 8)$ orientations for Gabor Wavelets, conventional Steerable Pyramid decomposition and our method.

Another important question that arises now, is to know how our feature classifier performs in both scaled datasets. To answer this question, we compared in Figures 4.13 and 4.14 the classification accuracies of the OPF against those obtained with SVMs. From Figure 4.13 we can see that by using a 16 dimension feature vector ($s = 2$ scales and $o = 4$ orientations) the OPF achieves a classification accuracy of 100%, which increases in turn the corresponding accuracy of the SVM up into two percent. Although this difference may appear despicable, note that the SVMs achieved the same accuracy when using a 28 dimension feature vector ($s = 2$ scales and $o = 7$ orientations). Thus, our recognition system requires almost only the half feature vector dimensionality to obtain a complete recognition. In the case of the *scaled dataset B*, the OPF achieved a 100% classification rate by using 24 ($s = 2$ scales and $o = 6$ orientations) dimension feature vectors. The SVMs achieved, in turn, this accuracy by using 30 ($s = 3$ scales and $o = 5$ orientations) dimension feature vectors. By considering again an 8 byte storage space for a float number, our recognizer uses $192(8 \times 24)$ bytes to classify texture images in an efficient manner, whereas by using SVMs $240(8 \times 30)$ bytes are needed.

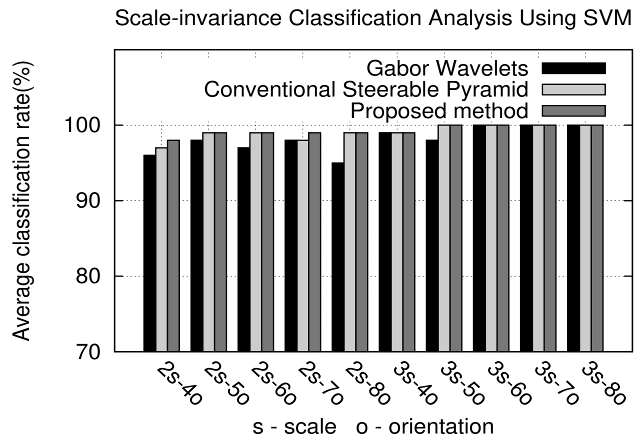


Figura 4.12: Classification accuracy comparison using the SVM classifier obtained in **scaled dataset B** using ($S = 2, 3$) scales with ($K = 4, 5, 6, 7, 8$) orientations for Gabor Wavelets, conventional Steerable Pyramid decomposition and our method.

4.6.3 Results Summarization

In this subsection, we provide a summary of our experimental results. For notation purposes, we will denote our image descriptor, the Gabor Wavelets, and the conventional Pyramid Decomposition descriptors as ID1, ID2, and ID3, respectively. The summarization of our results for the *rotated datasets* A and B is provided in Tables 4.1, and 4.2. Table 4.1, compares for each rotated dataset, the mean recognition rates obtained by the three texture image descriptors using different scales ($s = 2, 3$) and different orientations ($o = 4, 5, 6, 7, 8$). In this set of experiments, we used Gaussian-kernel Support Vector Machines (SVMs) as texture classification mechanisms. From our results, it can be noticed that our texture image descriptor performs better regardless of the dataset used, or the image decomposition parameters considered during feature extraction (number of scales and orientations). Furthermore, as it can be seen in Table 4.2, the OPF classifier improves the recognition accuracies obtained by the SVM classifier in all of our experiments. The summarized results for the *scale datasets* A and B are presented in Tables 4.3, and 4.4. As we can see, our proposed recognition system performs again better than the previously mentioned approaches in both feature extraction and classification tasks.

4.6.4 Training Sample Classification Rates

The achieved performances of our feature classifier using different number of training samples are shown graphically in Figures 4.15– 4.18. The y-axis denote the achieved average classification rate, whereas the x-axis represents the number of training samples

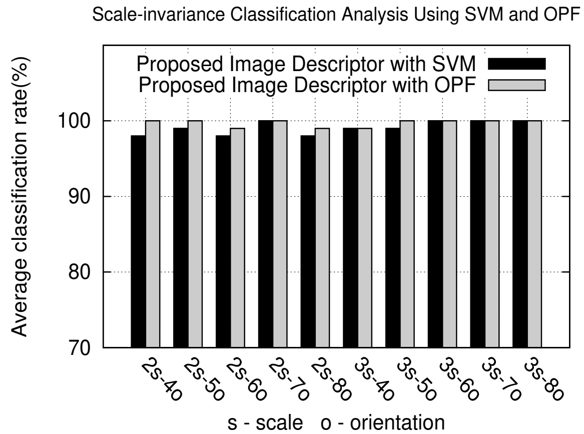


Figura 4.13: Classification accuracy comparison using the SVM classifier obtained in **scaled dataset A** using ($S = 2, 3$) scales with ($K = 4, 5, 6, 7, 8$) orientations for Gabor Wavelets, conventional Steerable Pyramid decomposition and our method.

Rotated Dataset	scales (s) / orientations (o)	ID1	ID2	ID3
A	(s=2;o=4,5,6,7,8)	**95.89%**	93.19%	93.19%
A	(s=3;o=4,5,6,7,8)	**97.99%**	92.36%	97.92%
B	(s=2;o=4,5,6,7,8)	**92.30%**	85.30%	91.29%
B	(s=3;o=4,5,6,7,8)	**96.70%**	85.76%	96.67%

Tabela 4.1: Mean recognition rates for the three different texture image descriptors using Gaussian-kernel Support Vector Machines as classifiers in the rotated datasets A and B.

considered. Each unique line belongs to each of the evaluated image descriptors (Gabor Wavelets, conventional Steerable Pyramid Decomposition, and our method). From those Figures we can see that almost all image descriptors attain reasonably good results even by using small dimentional feature vectors (85%+). However the superiority of our system can be clearly seen. Note that in the case of the *rotated datasets A* and *B* (Figures 4.15 and 4.16, respectively) our system remained with high accuracies above 97% and 95%, respectively. The analogous accuracies of Gabor Wavelets in both datasets have not reached the rates of our descriptor in any number of training samples used (en ninguna de las cantidades de muestras utilizadas). At the same time, our improvements over the conventional Steerable Pyramid Decomposition are notorious. In contrast, in the case of the *scale dataset A*, we can see that the accuracies of the image descriptors are very close to each other. However, our system achieved a 100% classification accuracy by using less training samples as the other two methods (Figure 4.17). Moreover, it can be seen from Figure 4.18 that by increasing the number of training samples, the conventional Steerable

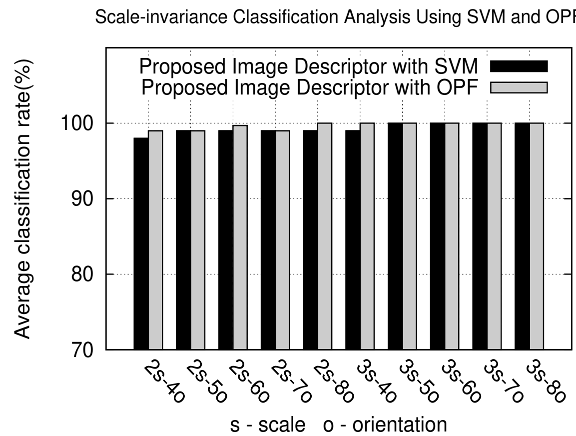


Figura 4.14: Classification accuracy comparison using the SVM classifier obtained in **scaled dataset B** using ($S = 2, 3$) scales with ($K = 4, 5, 6, 7, 8$) orientations for Gabor Wavelets, conventional Steerable Pyramid decomposition and our method.

Rotated Dataset	scales (s) / orientations (o)	OPF	SVM
A	(s=2;o=4,5,6,7,8)	**98.89%**	95.89%
A	(s=3;o=4,5,6,7,8)	**98.61%**	97.99%
B	(s=2;o=4,5,6,7,8)	**97.35%**	92.30%
B	(s=3;o=4,5,6,7,8)	**96.74%**	96.70%

Tabela 4.2: Mean recognition rates for the proposed rotation-invariant texture image descriptor using both OPF and SVM classifiers in the rotated datasets A and B.

Pyramid Decomposition does not improve much and in some cases it even deteriorates its classification accuracies. The curves in this Figure show that the average classification accuracies between our proposed image descriptor and the Gabor Wavelets are almost the same. However, the accuracies of our method are still higher. Finally, by analyzing all those results, we can see clearly that our method provides a significant improvement over the other approaches.

4.6.5 Recognition Processing Time

We also computed the recognition processing time for the classifiers in the evaluated datasets. Note that for computing the processing time, we considered both training and classification times together. Table 4.5 displays those values in seconds.

As we can see, the OPF algorithm is extremely faster than the SVM classifier. For the *rotated datasets A* and *B* as well as for the *scaled datasets A* and *B*, the OPF classifier

Scaled Dataset	scales (s) / orientations (o)	ID1	ID2	ID3
A	(s=2;o=4,5,6,7,8)	**98.78%**	93.19%	97.04%
A	(s=3;o=4,5,6,7,8)	**99.67%**	99.66%	96.05%
B	(s=2;o=4,5,6,7,8)	**99.35%**	97.12%	99.90%
B	(s=3;o=4,5,6,7,8)	**99.95%**	99.83%	99.44%

Tabela 4.3: Mean recognition rates for the three different texture image descriptors using Gaussian-kernel Support Vector Machines as classifiers in the scaled datasets A and B.

Scaled Dataset	scales (s) / orientations (o)	OPF	SVM
A	(s=2;o=4,5,6,7,8)	**99.03%**	98.78%
A	(s=3;o=4,5,6,7,8)	**99.89%**	99.67%
B	(s=2;o=4,5,6,7,8)	**99.58%**	99.35%
B	(s=3;o=4,5,6,7,8)	**100%**	99.95%

Tabela 4.4: Mean recognition rates for the proposed scale-invariant texture image descriptor using both OPF and SVM classifiers in the scaled datasets A and B.

was 112.15, 130.33, 125.69 and 126.30 times faster, respectively. The SVM algorithm had a slow performance due to the fact of the optimization procedure implemented in the libSVM [19]. However, by removing the optimization procedures, this processing time could be decreased. In turn, this could produce lower classification rates.

4.7 Conclusions

In this paper a novel texture classification system was proposed. Its main features include: (1) a new rotation-invariant and scale-invariant image descriptor, as well as (2) a novel multi-class recognition method based on Optimum Path Forest. The proposed image descriptor exploits the discriminatory properties of the Steerable Pyramid Decomposition for texture characterization. By finding either the **dominant orientation** or **dominant scale** value presented in the texture images, the feature elements are aligned according to this value. By doing this, a more reliable feature extraction process can be performed, since corresponding feature elements of distinct feature vectors, coincide with images either at same orientations or at same scales. In addition, our system adopted a novel approach for pattern classification based on Optimum Path Forest, which finds prototypes with none zero classification errors in the training sets and learns from errors in evaluation sets. By combining the discriminating power of our image descriptor

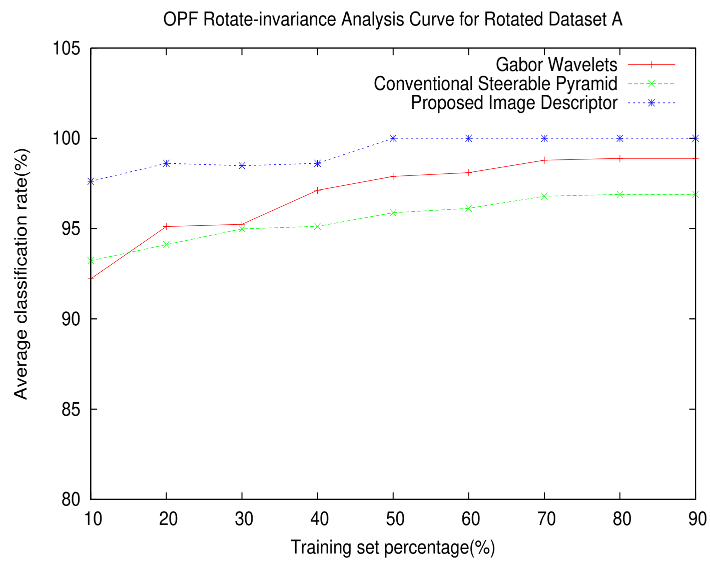


Figura 4.15: Average classification accuracy vs. number of training samples in **rotated dataset A**.

Dataset	OPF	SVM
Rotated Dataset A	**0.0260**	2.916
Rotated Dataset B	**0.0480**	6.256
Scaled Dataset A	**0.0388**	4.877
Scaled Dataset B	**0.0487**	6.151

Tabela 4.5: Execution times of the OPF and SVM approaches in seconds.

and classifier, our system uses small size feature vectors to characterize texture images without compromising overall classification rates, being ideally for real-time applications or for applications where data storage capacity is a limitation.

State-of-the-art results on four image datasets derived from the standard Brodatz database were further discussed. For the rotation-invariance evaluation, our method obtained a mean classification rate of 98.89% in comparision with a mean accuracy of 95.89% obtained by using SVMs in the *rotated dataset A*. In the case of the *rotated dataset B* those rates are 97.35% and 92.30%, respectively. Concerning the scale-invariance evaluation, our system improves classification rates from 98.78% to 99.03% in the case of the *scaled dataset A*, whereas in the *scaled dataset B* those rates are improved from 99.35% to 99.58%.

Further, the OPF multi-class classifier outperformed the SVM in the four datasets. It is a new promising graph tool for pattern recognition, which differs from traditional approaches, in that, it does not use the idea of feature space geometry, therefore, better

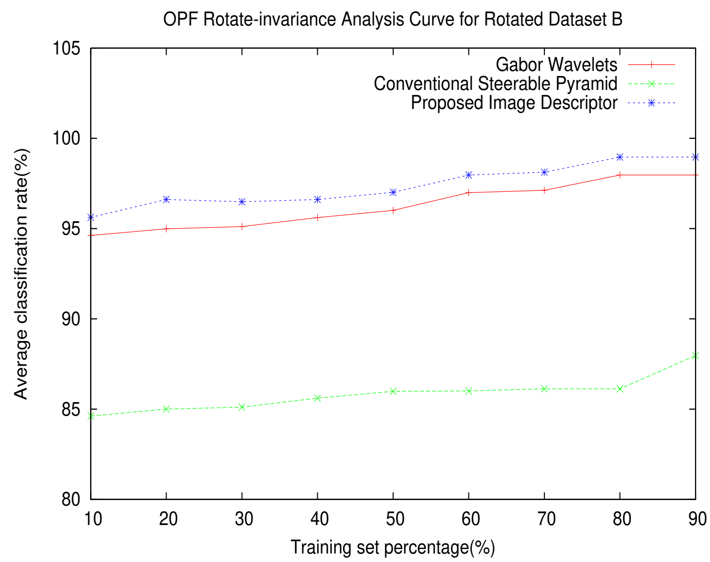


Figura 4.16: Average classification accuracy vs. number of training samples in **rotated dataset B**.

results in overlapped databases are achieved.

4.8 Acknowledgements

The authors would like to thank CNPq (Grants 302427/2004-0, 134990/2005-6, 477039/2006-5, and 311309/2006-2), Webmaps II CNPq project, FAPESP (Grant 03/14096-8), Microsoft Tablet PC Technology and Higher Education project, as well as CAPES/COFECUB (Grant 392/08) for their financial support. They would also like to thank the anonymous reviewers for their comments.

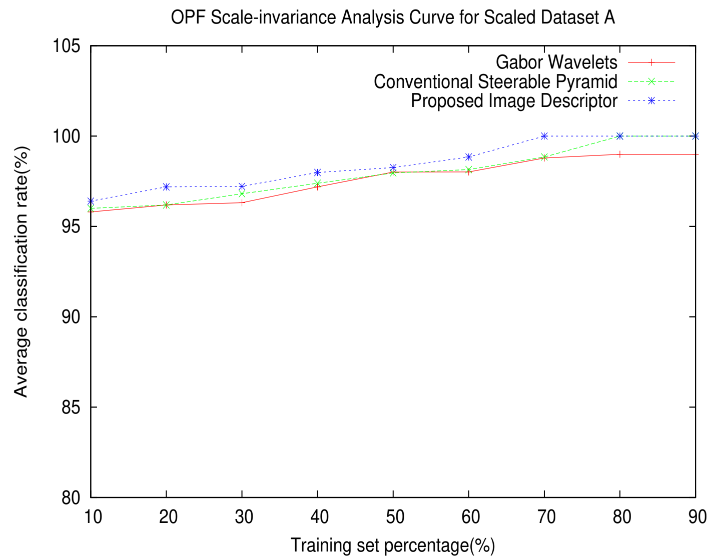


Figura 4.17: Average classification accuracy vs. number of training samples in **scaled dataset A**.

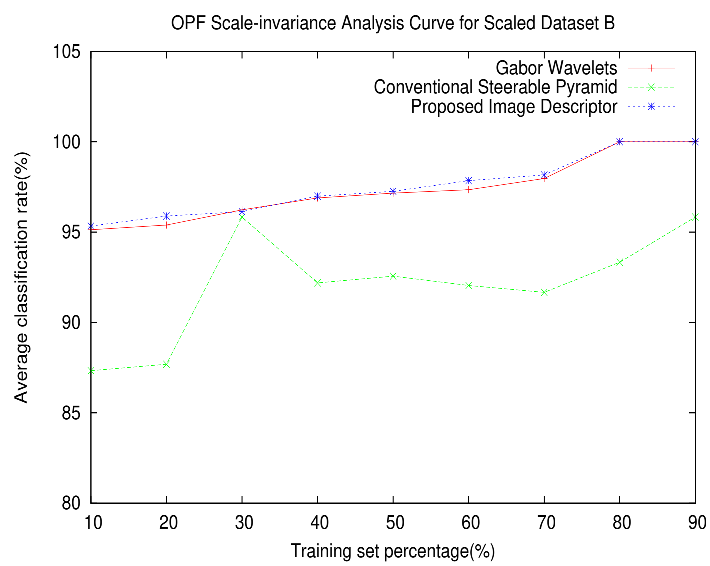


Figura 4.18: Average classification accuracy vs. number of training samples in **scaled dataset B**.

Capítulo 5

Wavelet-based fingerprint image retrieval

5.1 Introduction

Biometrics, which refers to the automatic recognition of individuals by their physical and/or behavioral characteristics, has emerged as a motivating and instigating research field [62]. In fact, several biometric applications have been adopted in civilian, commercial, and forensic areas. Traditionally, the physical characteristics used for human recognition include fingerprints [57], iris [144], retina [51], and face [150], while the behavioral ones include signature [99], voice [32], and gait [8]. Among all these biometric characteristics, fingerprints are considered one of the most reliable characteristic for human recognition due to their individuality and persistence [105]. The fingerprint's individuality means that it is unique across individuals and across fingers of the same individual, even in identical twins [61]. On the other hand, the fingerprint's persistence means that the basic fingerprint characteristics do not change with time.

The popularity of fingerprint-based recognition have led to the creation of large-scale databases. While the large size of these collections compromises the retrieval speed, the noise and the distortion that can be found in fingerprint images may reduce the overall retrieval accuracy. Therefore, both retrieval accuracy and speed play an important role in the fingerprint recognition process.

Automatic fingerprint recognition often involves four steps [56, 57]: (1) acquisition, (2) classification, (3) identification, and (4) verification. Fingerprint acquisition refers to the capture and representation of fingerprints. Fingerprint classification consists in assigning a fingerprint to a pre-defined class, whereas fingerprint identification is referred to the retrieval of fingerprint that corresponds to a given fingerprint query image (one-to-many comparisons). Finally, fingerprint verification is used to determine whether two fingerprint images are the same or not (one-to-one comparisons). However, considering the large size of fingerprint image databases and the computational cost of fingerprint verification algorithms, it is necessary to reduce the number of one-to-one comparisons during fingerprint verification, seeking both accuracy and retrieval speed.

In this sense, we propose an original approach to guide the search and the retrieval in fingerprint image databases. We study particularly the textural patterns that can be found in the central region of fingerprints in order to generate textural feature vectors used for fingerprint indexing and retrieval. For that purpose, we exploit the capability of different types of Wavelets transform to integrate both multiresolution and space-frequency properties in a natural manner. The fingerprint images are then decomposed into different spatial/frequency subimages and some statistical analysis is performed to generate feature vectors. These feature vectors are then used to compare the similarity among a given fingerprint query image and the database images.

In our approach, the texture features are extracted by different types of the Wavelet transform, which include: Steerable Wavelet (more known as Steerable Pyramid [37]), Gabor Wavelet transform (GWT [72, 87]), Tree-Structured Wavelet Decomposition using orthogonal filterbank (TSWD Haar, Daubechies 4-, 8-, and 16-Tap [27, 85]), and Tree-Structured Wavelets Decomposition using bi-orthogonal filterbank (Spline Wa-velets [137]).

In the case of the Steerable Pyramid, different orientations filters and decomposition levels are used to generate a translation- and rotation-invariant fingerprint representation. In the GWT, different scales and orientations are used to capture the relevant texture information, whereas in the TSWD, the image texture content is captured on the low frequency subband, while the high frequency subband is used to capture the image variations in different directions.

The focus of our work is to show the utility and suitability of applying different types of Wavelets transform for fingerprint indexing and retrieval. To the best of our knowledge, this work represents the first attempt to characterize the textural content of fingerprint images by using Wavelet transforms. In this context, the main contributions of this paper are:

1. A description of a texture-based retrieval system for fingerprint identification considering both feature extraction methods and similarity measures.

2. The treatment of the fingerprint identification problem as a wavelet-based image retrieval challenge.

3. A detailed comparison of the retrieval accuracy achieved by different combinations of types of Wavelet transforms and similarity measures in terms of precision and recall curves.

4. The validation of the suitability of the Wavelet transforms to represent and describe the fingerprint characteristics.

The remainder of this paper is organized as follows. The next section summarizes some related concepts, whereas section 5.3 presents an overview of related works and outlines the context of our study. Section 5.4 describes the architecture of the proposed system. The module used for detecting the center point region of the fingerprints is presented in section 5.5. Various wavelet-based feature extraction algorithms are described in section 5.6. The feature representation for each type of Wavelets is presented in Section 5.7. Different similarity metrics and the feature indexing method are described in section 5.8. The results of our experiments are discussed in section 5.9. Finally, our conclusions are drawn in section 5.10.

5.2 Background

In this section, we formalize the main terms used along this paper.

5.2.1 Image Descriptors

Definition 1. An **image descriptor** D is defined as a pair (ϵ_D, δ_D), where $\epsilon_D : I \longrightarrow \mathbb{R}^n$ is a function that extracts a feature vector $\vec{v_I}$ from a given image I, and $\delta_D : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}^n$ denotes the distance function used to compute similarity between two images considering their feature vectors. The smaller the distance, the more similar the images.

Definition 2. A **feature vector** $\vec{v_I}$ of an image I is a point in $\mathbb{R}^n$ space, such that: $\vec{v_I} = (v_1, v_2, \ldots, v_n)$, where n denotes the dimension of the vector.

5.2.2 Metric Spaces

Definition 3. A **metric distance** function $d()$ is a function that has the following properties:

(i) Symmetry: $d(O_1, O_2) = d(O_2, O_1)$

(ii) Positiveness: $0 < d(O_1, O_2) < \infty, \quad O_1 \neq O_2$ and $d(O_1, O_2) = 0$

(iii) Triangle inequality: $d(O_1, O_3) \leq d(O_1, O_2) + d(O_2, O_3)$

where O denotes the domain of a set of objects $O = (O_1, O_2, \ldots, O_n)$. The pair (O, d) is known as **metric space**. The similarity functions of the image descriptors are special cases of metric spaces.

Definition 4. A **Metric Access Method** (MAM) is a class of Access Method (AM) that is used to manage large volumes of metric data allowing insertions, deletions and searches [63].

5.2.3 Similarity Queries

In metric spaces two well-known similarity queries that can be performed are:

Definition 5. Given the query object O_q and the maximum search distance r_q, the **range query** $R_q(O_q, r_q)$ is used to retrieve all the objects $O_r \in O$ that satisfy the following condition: $d(O_q, O_r) < r_q$.

Definition 6. Given the query object O_q and the value $k \in Z^+$, the **k-Nearest Neighbor Query** ($kNN(O_q, k)$) retrieve the k-closest objects in $O_r \in O$ that satisfy the following properties: $|O_r| = k$ and $d(O_q, O_r) \leq d(O_q, O_s) \forall O_s \in O$.

5.3 Related Work

The most common way to reduce the number of one-to-many comparisons during fingerprint retrieval is to partitionate the database using fingerprint classification techniques, which can be divided into two main categories: exclusive and continuous classification.

The former uses information related to the pattern of ridges and valleys found in fingerprints to partitionate the fingerprint database into mutual exclusive bins. In this sense, once the fingerprint query image is classified, the image candidates are searched in the corresponding bin. Further, this kind of approach can be subdivided into four subcategories depending on the type of information used for exclusive classification, namely, ridge-, orientation field-, singularity-, and structural-based information. In continuous classification approaches, fingerprint images are represented by feature vectors. Similarities among fingerprint images are established by the distance in the feature space of their corresponding feature vectors. This approach is closely related to a fingerprint database indexing problem.

Ridge-based approaches use traditionally the information of the structure frequency of the fingerprint ridges for classification purposes. The work proposed by Fitz et al. [34] takes into account the frequency spectrum of fingerprints, obtained by applying a hexagonal Fourier Transform, to classify fingerprints into three classes: whorl, loops, and arches. A wedge-ring detector is used to partitioned the frequency domain images into non-overlapping areas where the pixel values are summed up to form a feature vector. Once the feature vector is found, it is compared to the reference feature vectors of each of the classes and further classification is performed by using a nearest neighbor classification method. To capture the structure of fingerprint ridges, some works develop mathematical models to characterize the fingerprint images [21, 59]. Chong et al. [21] use, for example, B-splines curves to approximate the shape of each of the fingerprint ridges. Then, similar orientation ridges are grouped together to obtain a global shape representation of the fingerprints, which is used for classification.

Approaches based on orientation field use the local average orientations of fingerprint ridges to classify fingerprints. Halici et al. [45] use the block orientation fields of fingerprints and certainty measures to generate the fingerprint feature vectors. For the sake of feature dimensionality reduction, they used a SOM neuronal network. Moreover, a second layer was added to the neuronal network architecture to improve the overall classification accuracy.

Fingerprint singularities have been widely used for classification [52, 65]. They can be defined as the local regions where the fingerprint ridges present some physical properties. Karu et al. [65] extract the singularities that can be found in the fingerprints to classify them by considering the location and the number of the detected singularities.

Structural approaches use the topology information of fingerprints for classification purposes. Maio et al. [82] segment the orientational field of fingerprint images to represent the fingerprints as relational graphs. For each class of fingerprints, a model relational graph is created. An inexact graph matching algorithm is used to classify fingerprint images.

Although the search spaces can be reduced in exclusive classification approaches, there are some shortcomings that should be considered: (1) some fingerprints present properties of more than one class and therefore they cannot be assigned into just one bin, (2) natural distribution of fingerprints is not uniform and therefore, even performing binning in the original database, the number of one-to-many comparisons can still be high – Cappelli et al. [17] proved that the distribution of fingerprint classes is not uniform (93.4% of fingerprints are among a set of three classes) –, and (3) some of the characteristics used for binning are not easy to detect due to the presence of noise, ambient conditions, etc.

On the other hand, Germain et al. [40] proposed a continuous system to index fingerprint databases using flash hashing. For that purposes, the location and orientation of the minutiaes, as well as the number of ridges among them are used to generate feature vectors. Some information related to the feature vectors are obtained and used to create the image indices that are added to a multi map memory structure and considered latter during fingerprint retrieval.

More recently, Tan et al. [131] compared two fingerprint identification approaches based on: (a) classification followed by verification, and (b) indexing followed by verification. Their classification approach uses Genetic Programming to generate compositor operators applied to some features extracted of the fingerprint orientation fields. A Bayesian classifier is them used to classify fingerprints. Their indexing approach is based on the work of Germain [40]. However, as a result of the retrieval process, a list of N fingerprint candidates is retrieved for the verification to determine the correspondence degree between the query image and the database images. They concluded that the indexing-based approach outperforms the classification considering the size of the search spaces.

Both approaches reduce the search spaces by considering some fingerprint singularities [40, 131]. Furthermore, the accurate detection of these singularities depend highly on the quality of the fingerprint images. Moreover, their definition often involves high computational costs, that will affect directly the fingerprint recognition time. On the other hand, they both use flash hashing for indexing purposes and we believe that by using metric access methods the query processing time will be improved. Thus, we will consider more specifically the textural information presented in fingerprints for feature extraction purposes, since they retain the discriminating power of fingerprints and Metric Access Methods (MAM) for their indexing. The description of the MAM used in our approach is beyond the scope of this paper, however a brief description is presented in section 5.8.

5.4 System Overview

The architecture of our proposed framework, presented in Figure 5.1, can be divided into two main subsystems, namely, the enrollment and the query subsystems. The enrollment subsystem acquires the information stored in the database for later use. The query subsystem is responsible for retrieving similar fingerprints from the database according to the user's fingerprint query image. Our system operates as follows:

1. Enrollment-subsystem: several fingerprint images are first captured (arrow labeled 1 in Figure 5.1) and a Region of Interest (ROI) within the fingerprint is marked (module 1, arrow 2) by a center point area detection module. A region of 64×64 pixels is used for marking the ROI. Feature extraction algorithms contained in the descriptor library (module B, arrow 3) are used by the feature extraction module to generate the features (arrow 4) that are indexed by a metric access method for further use.

2. Query-subsystem: a fingerprint query image is received as input from the user (arrow 1). Once the fingerprint ROI is detected (module A, arrow 2), the feature extraction algorithms contained in the descriptor library are used to extract the feature vectors from the query image (module B with arrows 3 and 4, respectively). The query image feature vector is used to rank the database images according to their similarity to the query image (module C). For that purpose, a distance computation algorithm

is selected from the descriptor library (arrow 5) and the metric access method is used to speed up the retrieval process (arrow 6). Finally, the most similar database images are ranked (arrow 7) and returned to the user (arrow 8).

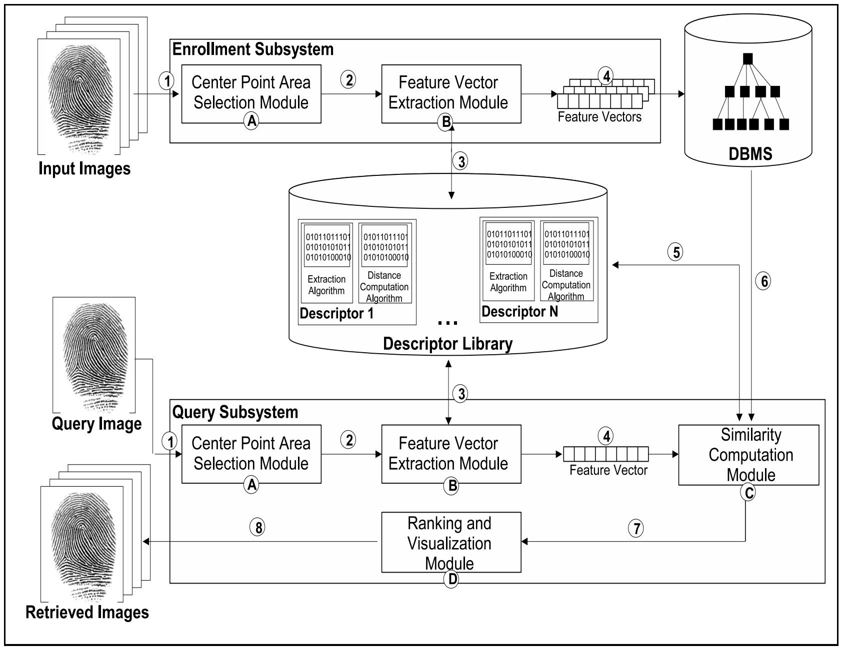


Figura 5.1: Architecture of the proposed system.

5.5 Center Point Area Detection

In order to detect the fingerprint center point area, we first locate the core point corresponding to the uppermost point contained in the inner-most ridge line. We have chosen the core point as the basis of the center area detection because: (1) it can be found in the central part of the fingerprint and, (2) appears more frequently than deltas [139]. The steps used for core detection are [60]:

1. Estimation and smoothing of the directional fields of the fingerprint input image.

2. Computation of the Poincaré index, in each (8×8) block. This index is defined as follows:

$$\text{Poincare(i,j)} = \frac{1}{2\pi} \sum_{k=0}^{N-1} \Delta(k) \tag{5.1}$$

$$\Delta(k) = \begin{cases} \delta(k) & \text{if } |\delta(k)| < \frac{\pi}{2} \\ \pi + \delta(k) & \text{if} \delta(k) \leq -\frac{\pi}{2} \\ \pi - \delta(k) & \text{otherwise} \end{cases} \tag{5.2}$$

$$\delta(k) = \theta(X(k^{'}), Y(k^{'})) - \theta(X(k), Y(k)) \tag{5.3}$$

where $k^{'} = (k+1) \bmod (N)$ and $\theta(i,j)$ is the directional field of the fingerprint image. $X(k)$ and $Y(k)$ are the coordinates of the blocks in the closed curve with N blocks. If the Poincaré index has a value of 1/2, then the current block is the core block. The center of this block is then the core point. If more than two cores are detected, go back to step 1 using a larger smoothing parameter for the directional fields.

Once the center point is obtained, a center point area can be easily defined. An image of size 64×64 pixels around the core point is then cropped, as shown in Figure 5.2.

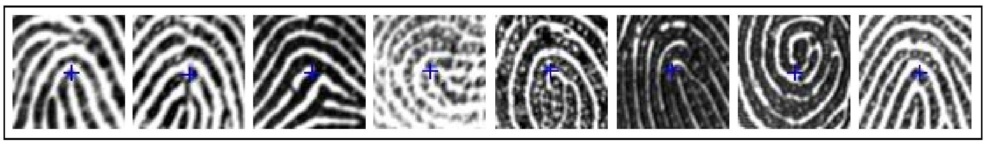


Figura 5.2: Some image samples of the center point area detected of 8 different fingerprints from FVC-2002 Database.

5.6 Feature Extraction

In this section, we provide a more detailed explanation of the various Wavelet-based feature functions used in the feature extraction module of the proposed architecture (See Module B in Figure 5.1). Wavelets have proved their efficiency in image retrieval problems due to their capability in capturing both texture and shape information [75, 100]. The Wavelet subband and multi-resolution decomposition are extremely adapted to compute relevant information of the data while preserving their basic content [22]. In this section, we present the wavelet-based feature extraction approaches considered in our system.

5.6.1 Texture Feature Extraction

The texture found in images represents a powerful discriminating feature for both image classification and retrieval. Although there does not exist a formal definition of texture, it can be understood as the basic primitives in images whose spatial distribution creates some visual patterns. Thus, the goal of a texture feature extraction method is to create a feature vector that captures the image texture information and preserves, at the same time, its content.

Considering that the ridges and valleys of fingerprints form a textural pattern, it is then possible to capture discriminatory information through their textural representations.

Further, the captured representations in the feature vectors are indexed and stored for image retrieval purposes.

5.6.2 Wavelet-based Feature Extraction

Here, the use of the Wavelet transform for texture description is motivated by two reasons [75]: (1) it integrates both multiresolution and space-frequency properties naturally, and (2) has demonstrated good accuracy for texture analysis and classification .

Wavelet Transform Review

The Wavelet Transform decomposes a signal $f(x)$ with a family of functions obtained through dilations and translations of a kernel function $\psi(x)$, called the mother wavelet. It is localized in both spatial and frequency domains. This family of functions is denoted by:

$$\psi_{m,n}(x) = 2^{\frac{-m}{2}}\ \psi(2^m x - n) \tag{5.4}$$

where $m, n \in \mathbb{Z}^+$ indicate dilations and translations, respectively. To construct the mother wavelet $\psi(x)$ a scaling function $\phi(x)$ is needed:

$$\phi(x) = \sqrt{2}\sum_k h(k)\ \phi(2x - k) \tag{5.5}$$

Then, the wavelet kernel $\psi(x)$ is determined as follows:

$$\psi(x) = \sqrt{2}\sum_k g(k)\ \phi(2x - k) \tag{5.6}$$

where:

$$g(k) = (-1)^k\ h(1 - k) \tag{5.7}$$

The explicit forms of $\phi(x)$ and $\psi(x)$ are not required to perform the wavelet transform, because it depends only on the coefficients $h(k)$ and $g(k)$ with low- and high-pass characteristics, respectively. The L-level decomposition of the signal $f(x)$ can be written as:

$$\begin{aligned} f(x) &= \sum_n c_{0,n}\ \phi_{0,n}(x) \\ f(x) &= \sum_n c_{L,n}\ \phi_{L,n}(x) + \sum_{l=1}^{L+1}\sum_n d_{l,n}\ \phi_{l,n}(k) \end{aligned} \tag{5.8}$$

the coefficients $c_{0,n}$ are given and the coefficients $c_{L,n}$ and $d_{l,n}$, both at scale l, are obtained by the coefficients $c_{l-1,n}$ at scale $l-1$ trough:

$$\begin{aligned} c_{l,n} &= \sum_k c_{l-1,n}\, h(k-2n) \\ d_{l,n} &= \sum_k c_{l-1,n}\, g(k-2n) \end{aligned} \tag{5.9}$$

where $1 \leq l \leq L+1$. A recursive wavelet decomposition can be obtained through $h(k)$ and $g(k)$ in Equation 5.9. The same process can be viewed as the convolution of signal $c_{l-1,n}$ with the impulse responses $\overline{h(n)} = h(-n)$ and $\overline{g(n)} = g(-n)$ of the low- and high-pass filters H and G, respectively (also known as quadrature filters), and then by downsampling the filtered signals by a factor of 2. The 2-D wavelet and scaling functions can be expressed as the tensor products of their 1-D complements:

$$\begin{aligned} \phi_{LL}(x,y) &= \phi(x)\phi(y) \qquad & \psi_{LH}(x,y) &= \phi(x)\psi(y) \\ \psi_{HL}(x,y) &= \psi(x)\phi(y) \qquad & \psi_{HH}(x,y) &= \psi(x)\psi(y) \end{aligned} \tag{5.10}$$

where $\phi_{LL}, \psi_{LH}, \psi_{HL}$, and ψ_{HH} represent the Low-Low, Low-High, High-Low, and High-High subbands, respectively.

Tree-Structured Wavelet Transform

In the Tree-Structured Wavelet Transform, the output of each of the subbands (ϕ_{LL}, ψ_{LH}, ψ_{HL}, ψ_{HH}) can be decomposed recursively. This decomposition is based on the fact that, for some kinds of textures, the most relevant information can be found in the middle subbands. To avoid a full decomposition, Chang et al. [20] proposed an energy-based criterion to decide which subband should be further decomposed. If the energy in a subband is very similar to the maximum energy of a subband at the same level, then further decomposition is not performed. However, for the sake of pattern retrieval, a fixed decomposition structure is convenient, since if facilitates distance computations and hence database browsing. On the other side, considering that the subband ψ_{HH} often leads to unstable features, recursively decomposition is done in the ϕ_{LL}, ψ_{LH}, and ψ_{HL} subbands.

For the Tree-Structured Wavelet Decomposition, we have considered two kinds of filterbanks, which include orthogonal and bi-orthogonal filterbanks. For the orthogonal filterbank we used the Haar Wavelets, and Daubechies 4-, 8-, and 16-tap [27,85]. For the bi-orthogonal filterbank the Spline wavelets is used [137]. Figure 5.3 presents an example of a two level Tree-Structured Wavelet Transform decomposition for image retrieval. Note that the ψ_{HH} subband is not further decomposed for the reasons explained above.

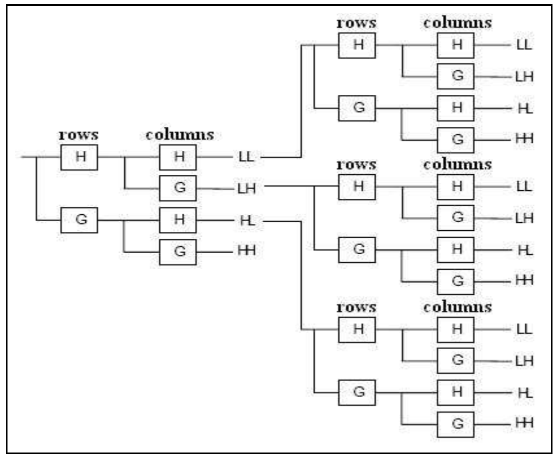


Figura 5.3: Two level Tree-Structured Wavelet Transform.

Gabor Wavelet Transform

A general 2-D Gabor function $\psi(x, y)$ is defined as:

$$\psi(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi jWx\right] \tag{5.11}$$

where the spatial coordinates (x, y) denote the centroid localization of the elliptical Gaussian window. The parameters σ_x and σ_y are the space constants of the Gaussian envelop along the x- and y-axes, respectively. The Fourier transform $G(u, v)$ of the Gabor function $\psi(x, y)$ can be written as:

$$G(u,v) = exp\left[\frac{-1}{2}\left(\frac{(u-W)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right)\right] \tag{5.12}$$

where W represents the frequency of the sinusoidal plane along the horizontal axis and the frequency components in x- and y-direction are denoted by the pair (u, v), while $\sigma_u = 1/2\pi\sigma_x$ and $\sigma_v = 1/2\pi\sigma_y$. Considering the non-orthogonal basis set formed by the Gabor functions, a localized frequency description can be obtained by expanding a signal with this basis.

Self-similar class functions, known as Gabor Wavelets, can be generated by dilations and rotations of the mother wavelet $\psi(x, y)$, i.e.:

$$\psi_{m,n}(x,y) = a^m\psi_{x',y'}, \quad a > 1 \tag{5.13}$$

considering $m = 1, \ldots S$ and $n = 1, \ldots K$. S and K denote the total number of dilations and orientations, respectively, and:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = a^{-m} \begin{bmatrix} cos\theta_n & sin\theta_n \\ -sin\theta_n & cos\theta_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{5.14}$$

where $\theta = n\pi/K$ and θ is the rotation angle. To ensure that the energy is independent of m, a scale factor a^{-m} is introduced. Considering the redundant information presented in the filtered images due to the non-orthogonality of the Gabor Wavelets, Manjunath et al. [87] designed a strategy to reduce the redundancy of the Gabor Wavelets Filterbank, where the half-peak magnitude of the filter responses touch each other in the frequency spectrum:

$$a = \left(\frac{U_h}{U_l}\right)^{\frac{1}{S-1}} \qquad \sigma_u = \frac{(a-1)U_h}{(a+1)\sqrt{2ln2}} \tag{5.15}$$

$$\sigma_v = tan\left(\frac{\pi}{2K}\right)\left[U_h - 2ln2\left(\frac{\sigma_u^2}{U_h}\right)\right]\left[2ln2 - \frac{(2ln2)^2\sigma_u^2}{U_h^2}\right]^{-\frac{1}{2}} \tag{5.16}$$

where $W = U_h$. The parameters U_h and U_l are used, respectively, to denote the upper and lower center frequencies of interest.

Steerable Pyramid

The Steerable pyramid is a linear multi-orientation and multi-scale image decomposition method, from which an image is subdivided into a collection of subbands localized at different scales and orientations [37]. This decomposition transform is based on convolution and decimating operations and has the advantage that the subbands are translation- and rotation-invariant. Using a high- and low-pass filter (H_0, L_0) the input image is initially decomposed into high- and low-pass subbands. The low-pass subband is further decomposed into a total of k-oriented band-pass portions $B_0, \ldots, B_k$ and into a lowpass subband L_1. The recursive decomposition is done by subsampling by a factor of 2 along the rows and columns the lower low-pass subband. The decomposition process of the first level of the Steerable Pyramid is shown in Figure 5.4. The white points in Figure 5.4 represent the resulting images after applying the directional filters on the input images, whereas the black point denotes the recursive pyramidal decomposition.

5.7 Feature Representation

The way the feature vectors are computed based on the different types of Wavelets presented in Section 5.6 is described in this section (See Arrow 4 in Figure 5.1). To generate the feature vectors, some statistical measures are used. More precisely, the mean μ_{mn} and the standard deviation σ_{mn} of the energy distribution of the multiresolution transform

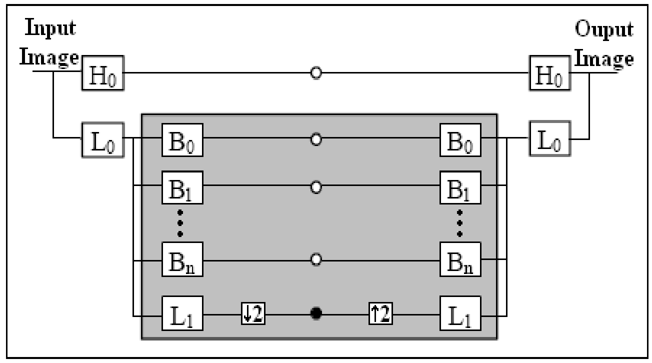


Figura 5.4: First Level of Steerable Pyramid Decomposition.

coefficients are used to capture the fingerprint textural information and thus to form the feature vector $\vec{f}$:

$$\mu_{mn} = \frac{1}{MN} \iint |W_{mn}(x,y)|\, dx\, dy \tag{5.17}$$

$$\sigma_{mn} = \sqrt{\iint \left(W_{mn}(x,y) - \mu_{mn}\right)^2 dx\, dy} \tag{5.18}$$

For the Tree-Structured Wavelet Transform the values of $|W_{mn}(x,y)|$ correspond to the energy distribution in one of the three subbands: LL, LH, and HL. Thus, the subindices m and n are integers that stand for the decomposition level and the current subband ($m = 1, 2, \ldots, L$ and $n = 1, 2, 3$), respectively. Furthermore, the 64×64 cropped image is decomposed into 6 levels, thus a total number of 18 subbands are considered. The feature vector $\vec{f}$ is formed as follows:

$$\vec{f_{TSWT}} = [\mu_{11}, \sigma_{11}, \mu_{12}, \sigma_{12}, \mu_{13}, \sigma_{13}; \ldots; \mu_{L1}, \sigma_{L1}, \mu_{L2}, \sigma_{L2}, \mu_{L3}, \sigma_{L3}] \tag{5.19}$$

For the Gabor Wavelet Transform, the values of $|W_{mn}(x,y)|$ denote the energy distribution of the transform coefficients after convolving an image I with the Gabor Wavelet $\psi_{m,n}$. Considering a total number of $S = 6$ scales and $K = 16$ orientations, the resulting feature vector is computed as follows:

$$\vec{f}_{GW} = [\mu_{11}, \sigma_{11}; \mu_{12}, \sigma_{12}; \ldots \mu_{S\,K}, \sigma_{S\,K}] \tag{5.20}$$

Considering $k = 16$ orientation subbands and $l = 6$ decomposition levels, the feature vector for the case of the Steerable Pyramid is generated as follows:

$$\vec{f_{SP}} = [\mu_{11}, \sigma_{11}, \mu_{21}, \sigma_{21}, \ldots, \mu_{k1}, \sigma_{k1}, \mu_{1l}, \sigma_{1l}, \mu_{2l}, \sigma_{2l}, \ldots, \mu_{kl}, \sigma_{kl}] \tag{5.21}$$

5.8 Feature Indexing

In this section, we present the different similarity measures studied in our work and the Metric Access Method used for feature indexing.

5.8.1 Similarity Measures

A key component of a CBIR (Content-based image retrieval) system are the similarity measure functions used for computing the similarity among images. This affirmation is valid if we consider that the retrieval performance depends not only on the effectiveness of the image features, but also on the similarity measures efficience. Thus, different similarity measures should be analyzed in a CBIR system to improve the retrieval performance. Let $\vec{x} = (x_1, x_2, \ldots, x_n)$ and $\vec{y} = (y_1, y_2, \ldots, y_n)$ be two feature vectors of dimension n, Table 5.1 presents the similarity measures studied in our work.

Evaluated Similarity Measures	
Measure	Formula
Bray Curtis	$d_{BC} = \sum_{i=1}^{n} \frac{\lvert x_i - y_i \rvert}{x_i + y_i}$
Canberra	$d_{C} = \sum_{i=1}^{n} \frac{\lvert x_i - y_i \rvert}{\lvert x_i \rvert + \lvert y_i \rvert}$
Euclidean	$d_E = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$
Manhattan	$d_M = \sqrt{\sum_{i=1}^{n} \lvert x_i - y_i \rvert}$
Squared Chord	$d_{SC} = \sum_{i=1}^{n} \left(\sqrt{x_i} - \sqrt{y_i}\right)^2$
Square Chi-Squared	$d_{SChi} = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{x_i + y_i}$

Tabela 5.1: Evaluated Similarity Measures.

5.8.2 Metric Access Method

To speed-up the retrieval, we have used a dynamic MAM known as Slim-tree [63]. The use of the Slim-tree in the fingerprint domain is attractive, since: (1) fingerprints can be inserted and deleted even after the creation of the tree, due to its dynamicity, (2) similarity queries such as kNN and range queries are supported and therefore CBIR applications are possible, (3) overlapping between nodes is minimized and thus the retrieval speed is increased and (4) due to its scalability, large amount of data can be handled in an efficient manner, even after having a grow the database. Furthermore, the Slim-tree has outperformed the well known M-tree indexing structure [106].

5.9 Experimental Results

In this section we present the experimental setup conducted in our study, as well as the effectiveness of the discussed feature extraction methods by means of some precision vs. recall curves.

5.9.1 Databases

Our experiments were conducted independently on two databases, the Bologna FVC-2002, and FVC-2004 databases, which consist of four different data sets, referred to as DB1, DB2, DB3, and DB4. Further, each of these data sets contains 8 fingerprint samples of 100 different fingers. Thus, each data set contains 880 fingerprints. Considering that each data set was collected by using different fingerprint technologies to cover better the advances in fingerprint sensing techniques, the size of each of the fingerprint images, as well as the resolution, vary among them. Data sets DB1, DB2, and DB3 were collected by different scanners technologies, whereas the data sets DB4 were collected by using a software for generating synthetic fingerprints [15,16]. Table 5.2 summarizes the different fingerprint technologies used to generate the databases.

These databases were not acquired in real environments and according to a formal protocol, therefore, the data is characterized by the presence of distortions (rotations, translations, low quality images) within fingerprints of the same individual's finger, being, therefore, useful for testing our system in extreme conditions.

5.9.2 Effectiveness Evaluation

In order to compare the retrieval effectiveness of different types of Wavelets, the proposed approach was measured in terms of Precision and Recall curves [43], since they have been widely used to evaluate retrieval effectiveness.

	FVC-2002			
	Technology	Scanner	Image Size	Resolution
DB1	Optical Sensor	Indentix TouchView II	388×374	500 dpi
DB2	Optical Sensor	Biometrika FX 2000	296×560	569 dpi
DB3	Capacitive Sensor	Precise Biometrics 100SC	300×300	500 dpi
DB4	Synthetic Software	SFingGE v2.51	288×384	About 500 dpi
	FVC-2004			
	Technology	Scanner	Image Size	Resolution
DB1	Optical Sensor	CrossMatch v300	640×480	500 dpi
DB2	Optical Sensor	Digital Persona U.are U.400	328×364	500 dpi
DB3	Thermal Sweeping Sensor	Atmel FingerChip	300×480	512 dpi
DB4	Synthetic Software	SFingGE v3.0	288×384	About 500 dpi

Tabela 5.2: Summarization of the FVC-2002, and FVC-2004 databases.

The precision is defined as the fraction of the retrieved images that are relevant to the given query, while the recall represents the proportion of relevant images among the retrieved ones. Thus, each of the retrieved images is considered as a match if it belongs to the same class as the query image. Considering the query image q and the number of correct, missed and false candidates (n_c, n_m, and n_f, respectively), the precision p_q in the first R retrieved images is defined as follows:

$$p_q = \frac{n_c}{n_c + n_f} = \frac{n_c}{R} \tag{5.22}$$

while the recall r_q of the such similar candidates S of the query image q is defined as:

$$r_q = \frac{n_c}{n_c + n_m} = \frac{n_c}{S} \tag{5.23}$$

5.9.3 Experiments

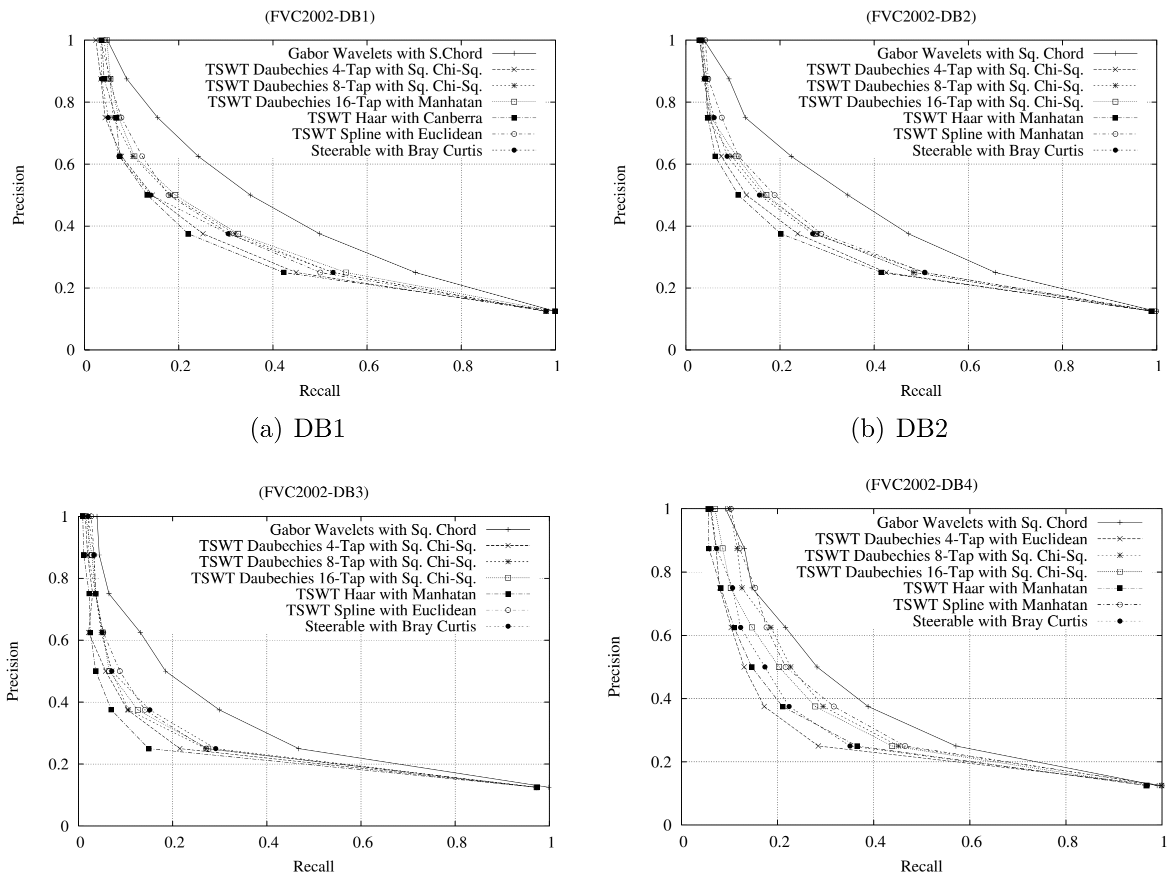
A series of experiments have been conducted to test the retrieval accuracy of our fingerprint image retrieval system using different wavelet-based feature extraction algorithms, and different similarity measures. The retrieval experiments were conducted independently in each of the four data sets of the two databases.

A simulated query is one of the 880 images in the data sets. Thus, a total number of 880 different fingerprint queries were performed per data set. The relevant retrieved images for each query are defined as the remaining fingerprints from the same individual, and the distances between the images are stored in increasing order.

A total number of 42 image descriptors were tested in each data set to determine the best pair of feature extraction algorithm with similarity measure. To generate them we have considered seven different wavelet-based feature extraction functions with the six similarity measures presented in Table 5.1. Seven different types of Wavelets including Gabor Wavelets, Steerable Pyramids, TSWD using Haar, Daub 4-tap, Daub 8-Tap, Daub 16-Tap, and Spline Wavelets have been evaluated.

Further, for each dataset we have considered the average retrieval effectiveness of all 42 image descriptors. Thus, a total number of $42 \times 2 \times 4$ experiments were conducted. The best combinations are then found, and the precision vs. recall curves showing the accuracy of the image descriptors are presented in figures 5.5, 5.6. Note that, for reasons of space, we report only the best results achieved by the image descriptors in each of the data sets.

Considering both figures, it is clear that the Gabor Wavelets outperform the other approaches in terms of retrieval accuracy. This is due to the fact that the Gabor Wavelets capture much useful information at different orientations when compared with the



(a) DB1 (b) DB2

(c) DB3 (d) DB4

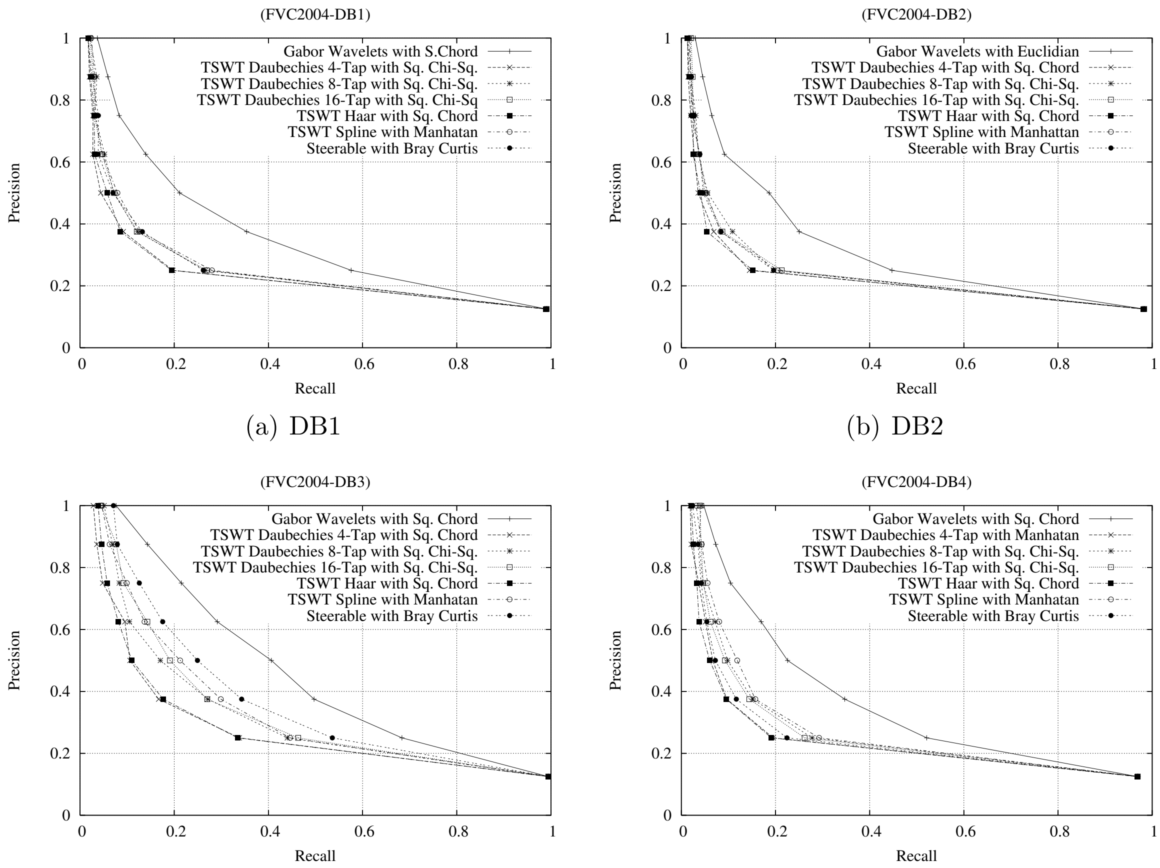
Figura 5.5: Average Precision vs. Recall curves for the FVC-2002 Database.

traditional Tree-Structured Wavelet Decompositions, which do not consider this specific information. In this sense, a more precise retrieval was performed due to the Gabor Wavelet's flexibility in controlling the orientation information.

In all eight data sets the best accuracy of Steerable Pyramids was achieved by using the same similarity measure, which is the Bray Curtis distance. This is because the difference on the numerator and the sume in the denominator normalizes the results, being the distance close to zero for very similar values, and close to one for very different values.

On the other side, we can observe that for each of the eight data sets, except for DB2 of the FVC-2004, the highest retrieval effectiveness was obtained by using the Square Chord similarity measure. The fact that the distance in each dimension is first obtained by performing the square root and then by applying the power of two, reduces the emphasis on those features with large dissimilarity.

The retrieval accuracy obtained by the different types of TSWD is almost the same.



(a) DB1 (b) DB2

(c) DB3 (d) DB4

Figura 5.6: Average Precision vs. Recall curves for the FVC-2004 Database.

The fact that this kind of Wavelet does not capture much information at different orientations has decreased their effectiveness in comparison with the Gabor Wavelets. The best similarity measure for the TSWD using Spline Wavelets are the Manhatan and Euclidian distances for both databases. That is because both of them, measure the absolute differences in each feature dimension in order to increase the retrieval effectiveness. However, one benefit of the Manhatan distance is that it requires less computational operations than the Euclidean distance.

In the case of the TSWD using Haar Wavelets, the best retrieval indices were obtained by using the Manhatan and Square Chord measures, whereas for the TSWD using Daubechies Wavelets, the best retrieval effectiveness was achieved by using the Square-Chi-Squared similarity measures. Note that, the best retrieval effectiveness of the different types of the TSWD was obtained by using the same similarity measures in both databases. This observation is important as the similarity measure used, should have the same effectiveness for self class images. In other words, these similarity measures have a better result for the domain of fingerprint images.

5.9.4 Image Retrieval Examples

Some retrieval examples are shown in Figure 5.7. The query image is on the top left corner, whereas the remaining ones are the retrieved images in increasing order of distance from the query image. Each fingerprint is labeled by the individual's ID and the number of the fingerprint sample, for example, image 22_3 will denote the individual number 22 and the third sample of his/her fingerprint. From Figure 5.7, it is possible to notice that most of the retrieved fingerprints are visually similar to the query image. This observation is important, because many of the errors on the retrieved data involve fingerprints that have the same texture and orientation as the query image, although they belong to other individuals, in other words, the presence of rotated or translated versions of similar fingerprints reduces the accuracy of the retrieval.

5.10 Conclusions

This paper has investigated the possibility of applying texture-based image retrieval techniques to reduce the search space for fingerprint identification. More specifically, we have proposed a novel approach to characterize fingerprint images by using different types of Wavelet transforms and similarity measures.

The retrieval effectiveness of the different image descriptors was compared by analyzing the results in terms of precision and recall. For all experiments, the best result was achieved by the Gabor Wavelet Transform combined with the Square Chord similarity measure. This fact relies basically on its flexibility to model the orientation and the scale information in images. Moreover, depending on the desired accuracy the descriptor parameter values can be adapted. The lack of this flexibility has influenced the retrieval performance of the Tree-Structured Wavelet Transform (TSWT), since it is not able to capture relevant information in different orientations.

It is important to notice that the databases used in our experiments do not reflect real acquisition conditions in the sense that image present abnormal distortions, including noise, significant rotations, and translations [83]. In this context, future work includes the use of databases containing more realistic fingerprint images. In this case, we expect that the retrieval effectiveness of our image descriptors will be considerably improved. In addition, we also plan to study the impact of using different metric access methods for fingerprint identification.

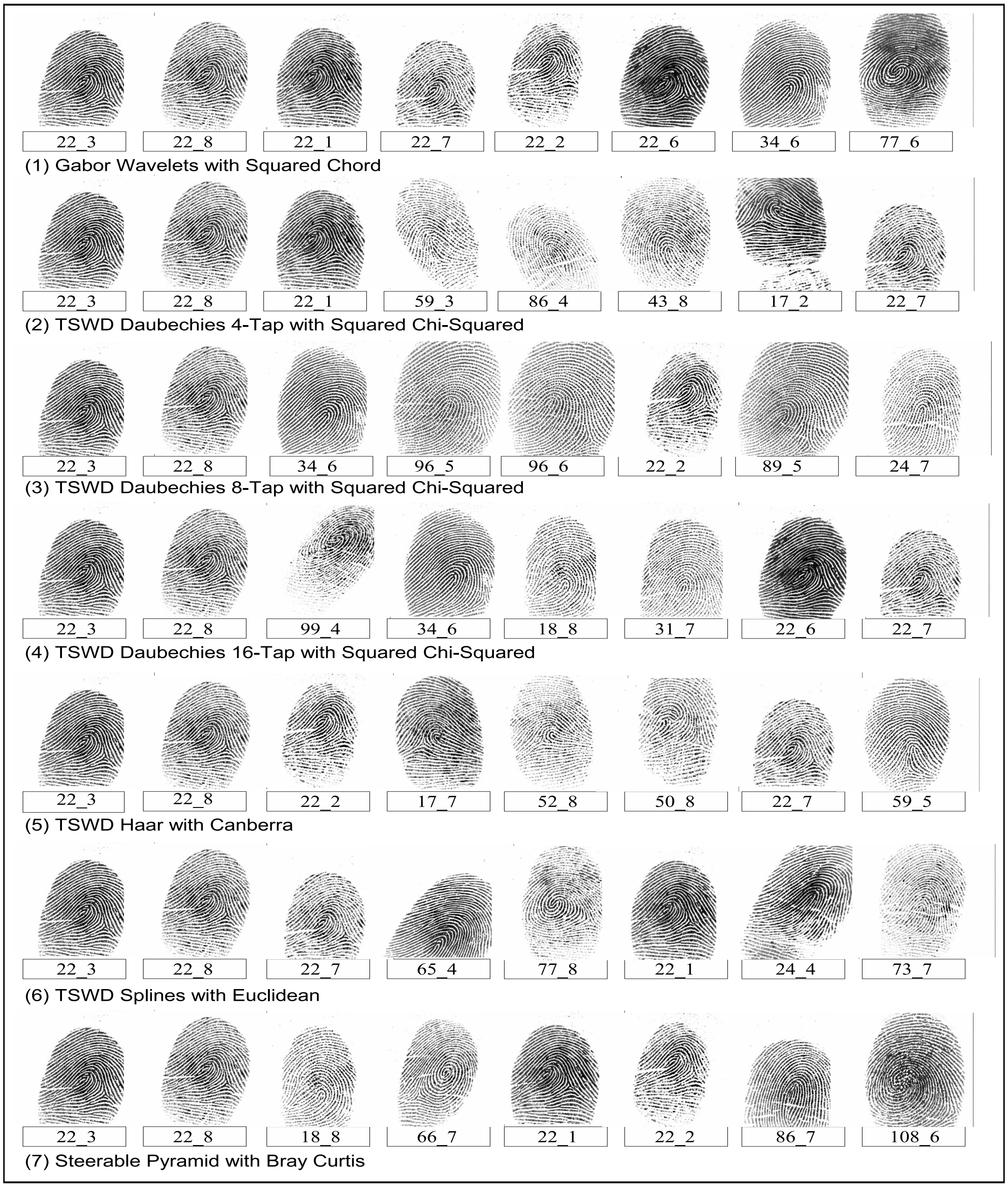


Figura 5.7: Some retrieval examples of the data set DB1 of the FVC-2002 database.

5.11 Acknowledgments

This work was partially financed by CNPq, CAPES, FAEPEX, FAPESP, and supported by Microsoft eScience grant.

Capítulo 6

Conclusões

Neste trabalho apresentou-se um estudo de técnicas de análise de textura para identificação e reconhecimento de imagens. Mais especificamente, foi proposto um novo descritor para caracterização de texturas que foi aplicado em tarefas de recuperação de imagens por conteúdo, bem como para reconhecimento de texturas.

A principal característica deste descritor consiste nas suas propriedades de representações texturais invariantes à rotação e escala. A nossa abordagem explora as propriedades discriminativas da Decomposição Piramidal *Steerable*.

O novo descritor leva em consideração a orientação ou a escala dominante presente nas imagens de textura. Devido às propriedades discriminativas da Decomposição Piramidal *Steerable*, os vetores de características possuem uma baixa dimensionalidade, possibilitando-se dessa maneira aplicações em que o espaço de armazenamento pode ser um fator importante.

Por outro lado, para analisar a acurácia do nosso descritor, no contexto de recuperação de imagens por conteúdo, um número total de 43056 consultas diferentes foram realizadas em três bases de imagens geradas a partir do padrão Brodatz. O descritor proposto apresentou boa eficácia nos experimentos realizados envolvendo bases de imagens que consideravam transformações de rotação e escala. Mais especificamente, as acurácias médias para as representações invariantes à rotação e escala foram incrementadas de 59.62% a 80.02% e de 80.5% a 86.51% em relação à Decomposição Piramidal Convencional, respectivamente. É importante mencionar que as taxas médias de recuperação em relação às *Wavelets* de Gabor foram menores sendo, respectivamente, de 61.42% e 60.7% para as bases de imagens.

Por outro lado, no contexto de reconhecimento de texturas, um novo sistema de classificação foi apresentado. Para tanto, utilizou-se o descritor proposto no presente trabalho, bem como o método de reconhecimento multiclasse baseado na abordagem *Optimum Path Forest* [108]. Combinando as propriedades do descritor e do classificador, o sistema proposto utiliza vetores de características de baixa dimensionalidade. Além disso, na avaliação das taxas de reconhecimento para invâriancia à rotação, o método obteve uma classificação média de 98.89% em comparação com a taxa média de 95.89% obtida pelas máquinas de vetores de suporte (*SVMs*) em uma das bases de imagens consideradas. Para uma outra base, as taxas foram respectivamente de 97.35% e 92.30% para ambos os métodos. Em relação à avaliação quanto à invariância à escala, o nosso sistema melhorou as taxas de classificação de 98.78% a 99.03% no caso de uma das bases de imagens com mudanças de escala, enquanto que para outra base estas taxas sofreram alterações de 99.35% para 99.58%.

Algumas possíveis extensões desse trabalho incluem:

1. Aplicação e análise do descritor para o problema específico de segmentação de texturas.

2. Validação do método para identificação e reconhecimento de imagens empregando outros tipos de *Wavelets* e classes de imagens.

3. Abordagem do uso dos descritores em problemas de indexação em bancos de imagens.

4. Validação do descritor considerando outras medidas de similaridade.

Referências Bibliográficas

[1] C. Allène, J. Y. Audibert, M. Couprie, J. Cousty, and R. Keriven. Some links between min-cuts, optimal spanning forests and watersheds. In *Mathematical Morphology and its Applications to Image and Signal Processing (ISMM'07)*, pages 253–264. MCT/INPE, 2007.

[2] Mohand Saïd Allili and Djemel Ziou. Globally adaptive region information for automatic color-texture image segmentation. *Pattern Recognition Letters*, 28(15):1946–1956, 2007.

[3] S. Arivazhagan, L. Ganesan, and S. Padam Priyal. Texture classification using gabor wavelets based rotation invariant features. *Pattern Recognition Letters*, 27:1976–1982, 2006.

[4] R. Audigier and R.A. Lotufo. Seed-relative segmentation robustness of watershed and fuzzy connectedness approaches. In *XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'07)*, pages 61–68, Belo Horizonte, MG, Oct 2007. IEEE CPS.

[5] R. Audigier and R.A. Lotufo. Watershed by image foresting transform, tie-zone, and theoretical relationship with other watershed definitions. In *Mathematical Morphology and its Applications to Signal and Image Processing (ISMM'07)*, pages 277–288, Rio de Janeiro, RJ, Oct 2007. MCT/INPE.

[6] Benjamin Balas. Attentive texture similarity as a categorization task: Comparing texture synthesis models. *Pattern Recognition*, 2007. Article in Press (Available online 31 August 2007).

[7] Benjamin Balas. Attentive texture similarity as a categorization task: Comparing texture synthesis models. *Pattern Recognition*, 2007. Article in Press (Available online 31 August 2007).

[8] Chiraz Benabdelkader, Ross G. Cutler, and Larry S. Davis. Gait recognition using image self-similarity. *EURASIP Journal on Applied Signal Processing*, 2004:572–585, 2004.

[9] S. Beucher and F. Meyer. The morphological approach to segmentation: The watershed. *Mathematical Morphology in Image Processing*, pages 433–481, 1993.

[10] A. Blake and C. Marinos. Shape from texture: Estimation, isotropy and moments. *Artifical Intelligence*, 45(3):323–380, 1990.

[11] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Workshop on Computational Learning Theory*, pages 144–152, 1992.

[12] Charles Bouman and Bede Liu. Multiple resolution segmentation of textured images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):99–113, 1991.

[13] A. C. Bovik, M. Clark, and W. S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73, 1990.

[14] D. Buckley, J.P. Frisby, and E. Spivey. Stereo and texture cue combination in ground planes: an investigation using the table stereometer. *Perception*, 20(91), 1991.

[15] R. Cappelli, D. Maio, and D. Maltoni. Synthetic fingerprint-database generation. In *Proceedings of the International Conference on Pattern Recognition (ICPR'02) Volume 3*, page 30744, Washington, DC, USA, 2002. IEEE Computer Society.

[16] R. Cappelli, D. Maio, D. Maltoni, and A. Erol. Synthetic fingerprint-image generation. *Proceedings of the International Conference on Pattern Recognition (ICPR'00)*, 03:3475, 2000.

[17] R. Cappelli, D. Maio, D. Maltoni, and L. Nanni. A two-stage fingerprint classification system. In *Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications (WBMA'03)*, pages 95–99, New York, NY, USA, 2003. ACM Press.

[18] Kenneth R. Castleman. *Digital Image Processing*. Prentice Hall, 1996.

[19] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at urlhttp://www.csie.ntu.edu.tw/ cjlin/libsvm.

[20] Tianhorng Chang and C.-C. Jay Kuo. Texture analysis and classification with tree-structured wavelet transform. *IEEE Transactions on Image Processing*, 2(4):429–441, October 1993.

[21] Chong, Ngee, Jun, and Gay. Geometric framework for fingerprint image classification. *Pattern Recognition*, 30(9):1475–1488, 1997.

[22] Charles K. Chui. *An introduction to wavelets*. Academic Press Professional, Inc., San Diego, CA, USA, 1992.

[23] M. Clerc and S. Mallat. The texture gradient equation for recovering shape from texture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):536–549, 2002.

[24] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT, 1990.

[25] J. Cousty, G. Bertrand, L. Najman, and M.Couprie. Watersheds, minimum spanning forests, and the drop of water principle. (IGM 2007-01), 2007. École Supérieure d'Ingénieurs.

[26] J. E. Cutting and R. T. Millard. Three gradients and the perception of flat and curved surfaces. *General Journal of experimental psychology*, 113(2):196–216, 1984.

[27] Ingrid Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36(5):961–1005, September 1990.

[28] Alberto del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann, 1st. edition, 1999.

[29] Yining Deng and b. s. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):800–810, 2001.

[30] Minh N. Do and Martin Vetterli. Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance. *IEEE Transactions on Image Processing*, 11(2):146–158, 2002.

[31] Richard C. Dubes and Anil K. Jain. Random field models in image analysis. *Journal of Applied Statistics*, 16(2):131–164, 1989.

[32] Anders Eriksson and Par Wretling. How flexible is the human voice? - a case study of mimicry. *Proceedings of EUROSPEECH*, 2:1043–1046, 1997.

[33] 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.

[34] Fitz and Green. Fingerprint classification using a hexagonal fast fourier transform. *Pattern Recognition*, 29(10):1587–1597, 1996.

[35] David A. Forsyth. Shape from texture without boundaries. In *Proceedings of the 7th European Conference on Computer Vision-Part III (ECCV'02)*, pages 225–239, London, UK, 2002. Springer-Verlag.

[36] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

[37] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[38] G.Cross and Anil K. Jain. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(1):25–39, 1983.

[39] Elisa Drelie Gelasca, Joriz De Guzman, Steffen Gauglitz, Pratim Ghosh, JieJun Xu, Emily Moxley, Amir M. Rahimi, Zhiqiang Bi, and B. S. Manjunath. Cortina: Searching a 10 million + images database. Technical report, September 2007.

[40] Robert S. Germain, Andrea Califano, and Scott Colville. Fingerprint matching using transformation parameter clustering. *IEEE Computational Science and Engineering*, 4(4):42–49, December 1997.

[41] Georgy L. Gimelf́arb and Anil K. Jain. On retrieving textured images from an image database. *Pattern Recognition*, 29(9):1461–1483, 1996.

[42] H. Greenspan, S. Belongie, R. Goodman, and P. Perona. Rotation invariant texture recognition using a steerable pyramid. *Proceedings of the 12th IAPR. International Conference on Pattern Recognition*, 2:162–167, 1994.

[43] William I. Grosky, Ramesh Jain, and Rajiv Mehrotra, editors. *Handbook of Multimedia Information Management.* Prentice-Hall, 1997.

[44] George M. Haley and B. S. Manjunath. Rotation-invariant texture classification using a complete space-frequency model. *IEEE Transactions on Image Processing*, 8(2):255–269, 1999.

[45] Halici and Ongun. Fingerprint classification through self-organizing feature maps modified to treat uncertainties. *Proceedings of the IEEE*, 84(10):1497–1512, 1996.

[46] Ju Han and Kai-Kuang Ma. Rotation-invariant and scale-invariant gabor features for texture image retrieval. *Image Vision Comput.*, 25(9):1474–1481, 2007.

[47] Robert M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, May 1979.

[48] J. K. Hawkins. Textural properties for pattern recognition. In *Picture Processing and Psychopictorics*, pages 128–136, New York, USA, 1970.

[49] J. K. Hawkins. Textural properties for pattern recognition. In B. C. Lipkin and A. Rosenfeld, editors, *Picture Processing and Psychopictorics*, pages 347–370, 1970.

[50] S. Haykin. *Neural networks: a comprehensive foundation.* Prentice Hall, 1994.

[51] Hill-99. *Retina Identification, Biometrics: Personal Identification in Networked Society.* Kluwer Academic, 1999.

[52] Lin Hong and Anil K. Jain. Classification of fingerprint images. In *11th Scandinavian Conference on Image Analysis, June 7-11, Kangerlussuaq, Greenland*, 1999.

[53] P. W. Huang, S. K. Dai, and P. L. Lin. Texture image retrieval and image segmentation using composite sub-band gradient vectors. *J. Visual Communication and Image Representation*, 17(5):947–957, 2006.

[54] Bernd Jähne. *Digital image processing.* Springer-Verlag, London, UK, fifth edition, 2002.

[55] Anil K. Jain and Farshid Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[56] Anil K. Jain, Lin Hong, and Ruud M. Bolle. On-line fingerprint verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):302–314, 1997.

[57] Anil K. Jain, Lin Hong, Sharath Pankanti, and Ruud Bolle. An identity-authentication system using fingerprints. *Proceedings of the IEEE*, 85(9):1365–1388, September 1997.

[58] Anil K. Jain and Kalle Karu. Learning texture discrimination masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):195–205, 1996.

[59] Anil K. Jain and Silviu Minut. Hierarchical kernel fitting for fingerprint classification and alignment. In *Proceedings of the International Conference on Pattern Recognition (ICPR'02)*, pages 469–473, 2002.

[60] Anil K. Jain, Salil Prabhakar, and Lin Hong. A multichannel approach to fingerprint classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):348–359, 1999.

[61] Anil K. Jain, Salil Prabhakar, and Sharath Pankanti. On the similarity of identical twin fingerprints. *Pattern Recognition*, 35(11):2653–2663, November 2002.

[62] Anil K. Jain, Arun Ross, and Salil Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):4–20, January 2004.

[63] Caetano Traina Jr., Agma Traina, C. Faloutsos, and B. Seeger. Fast indexing and visualization of metric data sets using slim-trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(2):244–260, 2002.

[64] B. Julesz. Texton gradients: The texton theory revisited. *Biological Cybernetics*, 54(4):211–251, 1986.

[65] Kalle Karu and Anil K. Jain. Fingerprint classification. *Pattern Recognition*, 29(3):223–231, 1999.

[66] Manesh Kokare, Prabir K. Biswas, and Biswanath N. Chatterji. Texture image retrieval using new rotated complex wavelet filters. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 35(6):1168–1178, 2005.

[67] Manesh Kokare, B. N. Chatterji, and P. K. Biswas. Cosine-modulated wavelet based texture features for content-based image retrieval. *Pattern Recognition Letters*, 25(4):391–398, 2004.

[68] Santhana Krishnamachari and Rama Chellappa. Multiresolution gauss-markov random field models for texture segmentation. *IEEE Transactions on Image Processing*, 6(2):251–267, 1997.

[69] Franci Lahajnar and Stanislav Kovačič. Rotation-invariant texture classification. *Pattern Recognition Letters*, 24(9-10):1151–1161, 2003.

[70] A. Laine and J. Fan. Texture classification by wavelet packet signatures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1186–1191, 1993.

[71] K.I. Laws. *Textured Image Segmentation.* PhD in Electrical Engineering, Department of Electrical Engineering – University of Southern California, 1980.

[72] Tai Sing Lee. Image representation using 2d gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):959–971, 1996.

[73] John-Peter Lewis. Texture synthesis for digital painting. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques (SIGGRAPH'84)*, pages 245–252, New York, NY, USA, 1984. ACM Press.

[74] Shutao Li, James T. Kwok, Hailong Zhu, and Yaonan Wang. Texture classi cation using the support vector machines. *Pattern Recognition*, 36(12):2883–2893, 2003.

[75] Kai-Chieh Liang and C.-C.J. Kuo. Waveguide: a joint wavelet-based image representation and description system. *IEEE Transactions on Image Processing*, 8(11):619–1629, November 1999.

[76] Ying Liu, Dengsheng Zhang, Guojun Lu, and Wei-Ying Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282, 2007.

[77] Ying Liu, Dengsheng Zhang, Guojun Lu, and Wei-Ying Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282, 2007.

[78] Anthony Lobay and D. A. Forsyth. Shape from texture without boundaries. *International Journal of Computer Vision*, 67(1):71–91, 2006.

[79] R.A. Lotufo and A.X. Falcão. The ordered queue and the optimality of the watershed approaches. In *Mathematical Morphology and its Applications to Image and Signal Processing (ISMM'00)*, volume 18, pages 341–350. Kluwer, Jun 2000.

[80] Wei-Ying Ma and B. S. Manjunath. A texture thesaurus for browsing large aerial photographs. *Journal of the American Society for Information Science*, 49(7):633–48, May 1998.

[81] Sebastian Magda and David Kriegman. Fast texture synthesis on arbitrary meshes. In *Proceedings of the 14th Eurographics workshop on Rendering (EGRW '03)*, pages 82–89, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[82] D. Maio and D. Maltoni. A structural approach to fingerprint classification. In *Proceedings of the International Conference on Pattern Recognition (ICPR '96) Volume*

III-Volume 7276, pages 578–585, Washington, DC, USA, 1996. IEEE Computer Society.

[83] Dario Maio, Davide Maltoni, Raffaele Cappelli, J. L. Wayman, and Anil K. Jain. Fvc 2002: Second fingerprint verification competition. *Proceedings of the International Conference on Pattern Recognition (ICPR'02)*, 3:811–814, 2002.

[84] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.

[85] Stephane G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[86] Varma Manik. *Statistical Approaches to Texture Classification*. PhD thesis, University of Oxford, oct 2004.

[87] Bangalore S. Manjunath and Wei-Ying Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.

[88] Ramchandra Manthalkar, P. K. Biswas, and B. N. Chatterji. Rotation invariant texture classification using even symmetric gabor filters. *Pattern Recognition Letters*, 24(12):2061–2068, 2003.

[89] Jianchang Mao and Anil K. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, 25(2):173–188, 1992.

[90] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004.

[91] Majid Mirmehdi and Maria Petrou. Segmentation of color textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2):142–159, 2000.

[92] Javier A. Montoya-Zegarra, Neucimar J. Leite, and Ricardo da S. Torres. Rotation-invariant and scale-invariant steerable pyramid decomposition for texture image retrieval. In *20th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'07)*, pages 187–193, 2007.

[93] Javier A. Montoya-Zegarra, Neucimar J. Leite, and Rircardo da S. Torres. Efficient and effective content-based image retrieval framework for fingerprint databases. In *V Workshop of Theses and Dissertations, XXI Brazilian Symposium on Databases*, pages 77–82, Florianopolis, Santa Catarina, Brazil, 16-20 October 2006.

[94] Javier A. Montoya-Zegarra, Neucimar J. Leite, and Rircardo da S. Torres. Multiresolution features for fingerprint image retrieval. In *Workshop of Theses and Dissertations, XIX Brazilian Symposium on Computer Graphics and Image Processing*, Manaus, Amazonas, Brazil, 8-11 October 2006.

[95] Javier A. Montoya-Zegarra, Neucimar J. Leite, and Rircardo da S. Torres. Wavelet-based Feature Extraction for Fingerprint Image Retrieval. Technical Report IC-06-12, Institute of Computing, State University of Campinas, September 2006.

[96] Javier A. Montoya-Zegarra, João P. Papa, Neucimar J. Leite, Ricardo da S. Torres, and A. X. Falcão. Rotation-invariant texture recognition. In *Advances in Visual Computing, Third International Symposium, ISVC 2007, Lake Tahoe, NV, USA, November 26-28, 2007, Proceedings*, volume 4842 of *Lecture Notes in Computer Science*, pages 193–204. Springer, 2007.

[97] Javier A. Montoya-Zegarra, João P. Papa, Neucimar J. Leite, Ricardo da S. Torres, and A. X. Falcão. Learning how to extract rotation-invariant and scale-invariant features from texture images. 2008. To appear in the Eurasip Journal of Advances in Signal Processing, Special Issue on Machine Learning.

[98] K. Muneeswaran, L. Ganesan, S. Arumugam, and K. Ruba Soundar. Texture classification with combined rotation and scale invariant wavelet features. *Pattern Recognition*, 38(10):1495–1506, 2005.

[99] Vishvjit S. Nalwa. Automatic on-line signature verification. *Proceedings of the IEEE*, 85(2):215–239, February 1997.

[100] Apostol Natsev, Rajeev Rastogi, and Kyuseok Shim. Walrus: A similarity retrieval algorithm for image databases. *IEEE Transactions on Knowledge and Data Engineering*, 16(3):301–316, 2004.

[101] S. Newsam, L. Wang, S. Bhagavathy, and B. S. Manjunath. Using texture to annotate remote sensed datasets. In *3rd International Symposium on Image and Signal Processing and Analysis (ISPA)*, Sep 2003.

[102] Fabrice Neyret and Marie-Paule Cani. Pattern-based texturing revisited. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*

(SIGGRAPH'99), pages 235–242, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[103] University of Southern California Signal and Image Processing Institute. Rotated textures. http://sipi.usc.edu/services/database/Database.html (Accessed on 1 March 2007).

[104] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[105] Sharath Pankanti, Salil Prabhakar, and Anil K. Jain. On the individuality of fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1010–1025, 2002.

[106] Marco Patella Paolo Ciaccia and Pavel Zezula. M-tree: An efficient access method for similarity search in metric spaces. *23rd International Conference on Very Large Data Bases (VLDB'97)*, pages 426–435, December 1997.

[107] J. P. Papa, A. X. Falcão, P. A. V. Miranda, C. T. N. Suzuki, and N. D. A. Mascarenhas. Design of robust pattern classifiers based on optimum-path forests. In *Proceedings of the 8th International Symposium on Mathematical Morphology (ISMM'07)*, 2007. accepted for publication.

[108] J. P. Papa, A. X. Falcão, P. A. V. Miranda, C. T. N. Suzuki, and N. D. A. Mascarenhas. A new pattern classifier based on optimum path forest. Technical Report IC-07-13, Institute of Computing, State University of Campinas, May 2007. Technical report available at `http://www.dcc.unicamp.br/ic-tr-ftp/2007/07-13.ps.gz`.

[109] J. P. Papa, A. X. Falcão, C. T. N. Suzuki, and N. D. A. Mascarenhas. A discrete approach for supervised pattern recognition. In *Proceedings of the 12th International Workshop on Combinatorial Image Analysis*, 2008. Accepted for publication.

[110] Alex Pentland, Rosalind W. Picard, and Stan Sclaroff. Photobook: Tools for content-based manipulation of image databases. *Proceedings of SPIE Conf. on Storage and Retrieval of Image and Video Databases*, 2(2):33–47, 1994.

[111] Alex P. Pentland. Fractal-based description of natural scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):661–674, November 1984.

[112] Rosalind W. Picard and Thomas P. Minka. Vision texture for annotation. *Multimedia Systems*, 3(1):3–14, 1995.

[113] R. M. Pickett. Visual analyses of texture in the detection and recognition of objects. In B. C. Lipkin and A. Rosenfeld, editors, *Picture Processing and Psychopictorics*, pages 289–308, 1970.

[114] Matti Pietikäinen, Timo Ojala, and Zelin Xu. Rotation-invariant texture classification using feature distributions. *Pattern Recognition*, 33(1):43–52, 2000.

[115] Javier Portilla and Eero P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal on Computer Vision*, 40(1):49–70, 2000.

[116] William K. Pratt. *Digital Image Processing: PIKS Inside*. John Wiley & Sons Inc, Los Altos, California, USA, third edition, 2001.

[117] Emil Praun, Adam Finkelstein, and Hugues Hoppe. Lapped textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques (SIGGRAPH'00)*, pages 465–470, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[118] Chi-Man Pun and Moon-Chuen Lee. Log-polar wavelet energy signatures for rotation and scale invariant texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):590–603, 2003.

[119] Till Quack, Ullrich Mönich, Lars Thiele, and B. S. Manjunath. Cortina: a system for large-scale, content-based web image retrieval. In *Proceedings of the 12th annual ACM international conference on Multimedia (MULTIMEDIA'04)*, pages 508–511, New York, NY, USA, 2004. ACM Press.

[120] Reed Tood R. and Hans Du Buf J.M. A review of recent texture segmentation and feature extraction techniques. *CVGIP: Image Understanding*, 57(3):359–372, May 1993.

[121] Trygve Randen and John H. Husoy. Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):291–310, 1999.

[122] P. K. Saha and J. K. Udupa. Relative fuzzy connectedness among multiple objects: Theory, algorithms, and applications in image segmentation. *Computer Vision and Image Understanding*, 82(1):42–56, 2001.

[123] Nicu Sebe and Michael S. Lew. Texture features for content-based retrieval. In *Principles of Visual Information Retrieval*, chapter 3, pages 51–86. Springer-Verlag, New York, 2001.

[124] E. P. Simoncelli and William T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. *Proceedings of the IEEE International Conference on Image Processsing (ICIP'95)*, 13(9):891–906, 1995.

[125] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[126] John R. Smith and Shih-Fu Chang. Visualseek: a fully automated content-based image query system. In *Proceedings of the 4th ACM international conference on Multimedia (MULTIMEDIA'96)*, pages 87–98, New York, NY, USA, 1996. ACM Press.

[127] Baris Sumengen and B. S. Manjunath. Graph partitioning active contours (gpac) for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):509, 2006.

[128] Todd James T. and Akerstrom Robin A. Perception of three-dimensional form from patterns of optical texture. *Journal of Experimental Psychology: Human Perception and Performance.*, 13(2):242–255, 1987.

[129] H. Tamura, S. Mori, and Y. Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 8:460–473, June 1978.

[130] T. N. Tan. Rotation invariant texture features and their use in automatic script identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):751–756, 1998.

[131] Xuejun Tan, Bir Bhanu, and YingQiang Lin. Fingerprint identification: classification vs. indexing. *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 151–156, July 2003.

[132] James T. Todd and Augustinus H.J. Oomes. Generic and non-generic conditions for the perception of surface shape from texture. *Vision Research*, 42(7):837–850, 2002.

[133] M. Tüceryan and Anil K. Jain. Texture segmentation using voronoi polygons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):211–216, 1990.

[134] Greg Turk. Texture synthesis on surfaces. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH'01)*, pages 347–354, New York, NY, USA, 2001. ACM Press.

[135] M. Unser and M. Eden. Multiresolution feature extraction and selection for texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):717–728, 1989.

[136] Michael Unser. Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing*, 4(11):1549–1560, 1995.

[137] Michael Unser, Akram Aldroubi, and Murray Eden. A family of polynomial spline wavelet transforms. *Signal Processing*, 30(2):141–162, 1993.

[138] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.

[139] Furong Wang, Xuan Zou, Yuan Luo, and Jiaxi Hu. A hierarchy approach for singular point detection in fingerprint images. *Lecture Notes in Computer Science*, 3072:359–365, January 2004.

[140] Lei Wang and B. S. Manjunath. A semantic representation for image retrieval. In *Proceedings of the International Conference on Image Processing (ICIP'03))*, pages 523–526, 2003.

[141] Yuzhong Wang, Jie Yang, and Ningsong Peng. Unsupervised color-texture segmentation based on soft criterion with adaptive mean-shift clustering. *Pattern Recognition Letters*, 27(5):386–392, 2006.

[142] Li-Yi Wei. *Texture synthesis by fixed neighborhood searching*. PhD thesis, Stanford, CA, USA, 2002. Adviser-Marc Levoy.

[143] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques (SIGGRAPH'00)*, pages 479–488, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[144] Richard P. Wildes. Iris recognition: An emerging biometric technology. *Proceedings of the IEEE*, 85(9):1348–1363, September 1997.

[145] Jiahua Wu. *Rotation Invariant Classification of 3D Surface Texture Using Photometric Stereo*. PhD in Computer Science, Department of Computer Science – School of Mathematical and Computer Sciences – Heriot-Watt University Edinburgh, 2003.

[146] Jiahua Wu and Mike J. Chantler. Combining gradient and albedo data for rotation invariant classification of 3d surface texture. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, pages 848–855, Washington, DC, USA, 2003. IEEE Computer Society.

[147] Fei Wua, Changshui Zhangb, and Jingrui He. An evolutionary system for near-regular texture synthesis. *Pattern Recognition*, 40(8):2271–2282, 2007.

[148] J. P. M. Zegarra, N. J. Leite, and R. da S. Torres. Wavelet-based Feature Extraction for Fingerprint Image Retrieval. *Journal of Computational and Applied Mathematics*, 2008. Accepted.

[149] Dengsheng Zhang, Aylwin Wong, Maria Indrawan, and Guojun Lu. Content based image retrieval using gabor texture features. In *Proceedings of 1st IEEE Pacific-Rim Conference on Multimedia (PCM'00)*, pages 392–395, 2000.

[150] Wei Zhao, Rama Chellappa, P. Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, December 2003.