

Algoritmo de Detecção de Formas de Interesse em imagens digitais para uma plataforma distribuída

Anderson H. S. Marcondes¹, Maurício A. Pillon¹, Alexandre G. Silva¹

¹Departamento de Ciência da Computação
Universidade do Estado de Santa Catarina (UDESC)
Campus Universitário Prof. Avelino Marcante s/n
89223-100 – Joinville – SC – Brasil

anderson_marcondes@yahoo.com.br, {mpillon, alexandre}@joinville.udesc.br

Abstract. *The technological advance and the popularization of computers enabled the society aggregates services to their daily lives. Nowadays, the relationship of people with machines is inevitable, such as medical equipments for MRI. In this context, this work aims to investigate techniques for different areas of computing that can be used in complex routine services based on analysis of images (image processing) efficiently (distributed systems). It was observed that limitations in the area of image processing can be provide by techniques of distributed systems. This paper proposes an algorithm of distribution for an application of detecting shapes of interest in digital images, and presents the preliminary results.*

Resumo. *O avanço tecnológico e a conseqüente popularização de computadores permitiram a sociedade agregar serviços ao seu cotidiano. Hoje em dia, o relacionamento de pessoas com máquinas é inevitável como, por exemplo, equipamentos médicos de ressonância magnética. Neste contexto, este trabalho tem por objetivo geral investigar técnicas de áreas distintas da computação que possam ser utilizadas em complexos serviços rotineiros baseados na análise de imagens (processamento de imagens) de forma eficiente (sistemas distribuídos). Observou-se que limitações da área de processamento de imagens podem ser supridas por técnicas da área de sistemas distribuídos. Este artigo propõe um algoritmo de distribuição para uma aplicação de detecção de formas de interesse em imagens digitais, e apresenta os resultados preliminares obtidos.*

1. Introdução

A computação, vista como atividade de meio, tem por objetivo auxiliar na resolução de problemas. As categorias de problemas tratados estão relacionadas com as mais diversas ciências, como física, biologia, química, geologia, astronomia, economia, artes, apenas para citar alguns exemplos. O avanço tecnológico, tanto de hardware quanto de software, destas últimas décadas, possibilitou a popularização de equipamentos e aplicações. A presença de computação, visível ou não (ubíqua/pervasiva), está por todos lugares: restaurantes, hospitais, residências. Neste contexto, destacam-se duas áreas importantes: processamento de imagens e sistemas distribuídos.

Atualmente, o reconhecimento de imagens é essencial a algumas áreas, tais como a médica, onde aplicações de auxílio a diagnósticos são dependentes de algoritmos de

detecção e filtros de imagens. A correta identificação de uma anomalia em um paciente pode ser decorrente da qualidade (detalhamento) da imagem oferecida ao médico após etapas computacionais realizadas internamente pelos equipamentos de ressonância magnética ou de tomografia. Alguns fatores que influenciam, de forma geral, no processamento de imagens são a qualidade de aquisição, o tratamento e a detecção de formas. Uma das restrições impostas aos algoritmos para manipulação de imagens é o poder computacional dos computadores. A necessidade de mais recursos para executar estas aplicações, visualizar e, principalmente, analisar as imagens é um problema eminente nesta área [Ostermann 2002].

Preocupações com o melhor aproveitamento das potencialidades dos equipamentos computacionais, técnicas de interconexões, dos métodos e dos modelos de comunicação são algumas das atividades de interesse das áreas de redes de computadores e sistemas distribuídos. Focando-se em aplicações específicas, pode-se ressaltar a problemática enfrentada pela área de processamento de imagens, no exemplo de aplicações médicas, no que tange a necessidade de poder computacional. A relação entre a carência da área de processamento de imagens e o foco de interesse de sistemas distribuídos é direta quando se trata de poder computacional [Coulouris et al. 2005]. Constatado este vínculo, este trabalho identificou técnicas de sistemas distribuídos apropriadas a aplicações de detecção de formas de interesse (processamento de imagens).

Alguns trabalhos comuns as estas duas áreas já vêm sendo estudados há algum tempo. Aplicações de classificação de imagens em rede de estações de trabalho [Subramanya 1999], processamento de imagens em sistema de memória distribuída [Nicolescu and Jonker 2000], ou, ainda, sistema distribuído para compreensão integrada de imagem com reconhecimento de imagens [Fink et al. 1996] são alguns exemplos. No entanto, nenhum dos trabalhos correlatos citados trata a questão de detecção de formas de interesse distribuída com foco na divisão de uma estrutura de representação de imagens denominada *max-tree*. Espera-se que esta abordagem forneça duas contribuições diretas à aplicação: redução do tempo de detecção e aumento da capacidade de processamento de imagens com dimensões maiores.

Neste artigo serão apresentados: (a) a arquitetura do algoritmo de detecção de formas de interesse distribuído baseado na *max-tree*; (b) resultados obtidos com a implementação deste e a execução em uma plataforma distribuída. Este texto está organizado em cinco seções: (1) a introdução que visa contextualizar as duas áreas de abrangência do trabalho; (2) a descrição do algoritmo de detecção de formas de interesse original em seqüencial; (3) a arquitetura distribuída do algoritmo de detecção de formas de interesse; (4) a plataforma de teste e a discussão sobre os resultados obtidos; (5) uma breve conclusão com trabalhos futuros.

2. Algoritmo de Detecção de Formas de Interesse

O Algoritmo de Detecção de Formas de Interesse se utiliza da representação de imagens através de uma estrutura hierárquica, *max-tree*, baseada em regiões [Cassol 2005], [Silva and de Alencar Lotufo 2007], [Silva et al. 2006]. A principal limitação no que se refere ao processamento de imagens do algoritmo seqüencial é o tempo de resposta impraticável para imagens com grandes dimensões, problema que será tratado ao longo deste trabalho. □

Para a representação de uma estrutura hierárquica de regiões utilizou-se o modelo *max-tree* que apresenta a representação de informações semânticas a partir do conjunto de componentes de nível da imagem [Salembier et al. 1998]. Esta representação caracteriza detalhes que esta estrutura possui através de uma série de atributos que identificam cada região, que são diferenciadas de acordo com a sua forma. As principais características da *max-tree* são: (1) ela é composta por nós, cada um representando um componente para um certo limiar - valor usualmente entre 0 e 255 (1 byte) que é atribuído aos *pixels*, podendo assim diferenciar as várias partes de uma região da imagem [Facon 2002] - de nível de cinza; (2) cada componente de nível representa uma única região da imagem; (3) cada região é determinada de acordo com um limiar de nível de cinza – cada limiar pode gerar múltiplas regiões - e um rótulo identificador ; (4) as folhas representam os máximos regionais da imagem - uma região onde as regiões ao redor tem nível de cinza menor - ; (5) a raiz representa o limiar de menor tom de cinza (região que contém todos os pixels da imagem).

O algoritmo *max-tree* baseia-se em inundação recursiva [Salembier et al. 1998]. A evolução desta inundação se dá a partir do menor nível de cinza (nó raiz da árvore) até seus nós folhas (máximos regionais). Ao final desta inundação tem-se uma árvore contendo as regiões que representa a *max-tree*. A representação da imagem mapeada à *max-tree* é ilustrada pela Figura 1, onde em (a) tem-se a imagem original com suas figuras em diferentes tons de cinza; (b) a árvore de regiões conforme o tom de cinza associado; (c) a *max-tree*, árvore hierárquica de regiões.

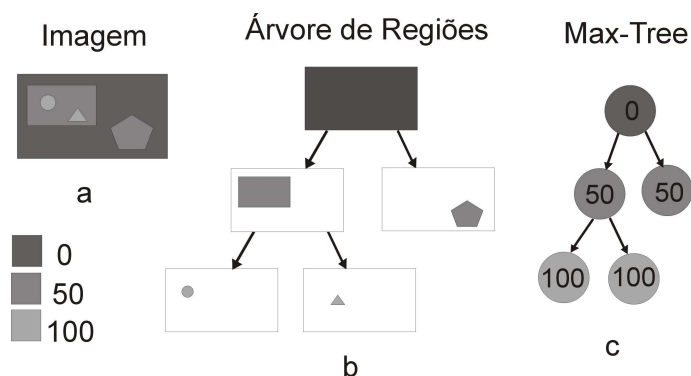


Figura 1. Representação da *max-tree*. (a) imagem de entrada; (b) representação da imagem dividida em regiões; (c) árvore hierárquica de regiões.

O algoritmo de detecção de formas baseia-se na identificação de regiões de interesse para cada um dos nós da árvore. Entende-se por formas, uma linha, uma reta, um círculo, uma elipse, ou outro lugar geométrico qualquer, ou forma determinada por um template (modelo). A função de análise de forma é um parâmetro do algoritmo de detecção. Ele procede uma busca em profundidade na *max-tree* e, para cada nó visitado, uma análise de forma é aplicada. Caso a forma seja identificada, o nó é preservado, se não, ele é removido da estrutura, e uma imagem resultante é gerada com as regiões dos nós preservados [Silva and de Alencar Lotufo 2007].

O reconhecimento de formas acontece seguindo etapas que são formadas pela imagem, representada pela *max-tree*, nó e região. As quatro etapas necessárias (Figura 2) ao reconhecimento são: (1) **representação**, determinação da *max-tree* da imagem de entrada; (2) **percurso**, resultando a seleção dos nós da *max-tree* com atributos de interesse; □

(3) **reconstrução** da região de cada nó selecionado; (4) **reconhecimento** de cada região por algoritmos específicos que detectam formas.

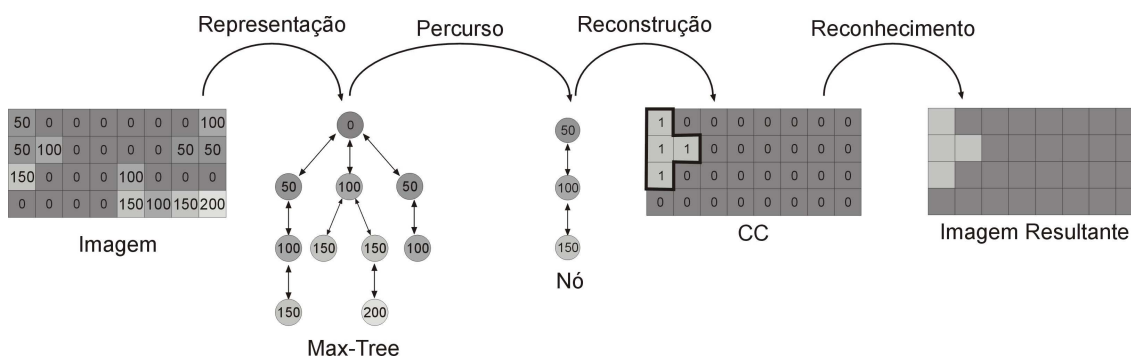


Figura 2. O Reconhecimento de formas passa por 4 etapas: (1) representação; (2) percurso; (3) reconstrução e (4) reconhecimento.

A aplicação das técnicas de processamento de imagens citadas permite a detecção de formas através da representação da imagem, *max-tree*. No entanto, o uso desta aplicação de reconhecimento de imagem ainda é restrito devido à limitação referente ao tempo de processamento deste algoritmo em seqüencial. Neste contexto, surgiu a proposta de uma arquitetura de distribuição para este algoritmo de reconhecimento de formas.

3. Algoritmo de Detecção de Formas de Interesse Distribuído

A definição da arquitetura do algoritmo de detecção de formas de interesse distribuído baseou-se na análise das atividades do algoritmo seqüencial. Identificou-se quatro etapas distintas ao reconhecimento da imagem. São elas: (1) entrada da imagem; (2) construção da *max-tree* – representação da imagem em uma árvore de regiões; (3) detecção da forma através da execução do algoritmo de detecção escolhido; e (4) geração da imagem resultante. A etapa (3), "detecta forma", é aplicada a cada região da *max-tree*. Ou seja, o mesmo algoritmo de detecção é executado sobre diferentes regiões (independentes) diversas vezes.

Em seguida, buscou-se identificar a etapa do algoritmo com maior custo computacional. Assim, pode-se modelar um grafo de dependência entre os procedimentos a fim de verificar a possibilidade de distribuição da(s) etapa(s) em questão. Com este estudo foi possível comprovar que as etapas (1) e (4) têm um custo computacional desprezível, enquanto o custo da etapa (2) varia de 5% a 10% do custo computacional e a etapa (3) consome mais de 90% (em alguns casos) do computacional gasto pela aplicação [Vieira 2008].

A detecção de forma de interesse, como apresentado na seção 2 (Figura 2), consiste na aplicação de um algoritmo com a lógica que descreve o lugar geométrico (ou com uma imagem de template) escolhido a ser aplicado (ou comparado) sobre cada região da imagem. Lembrando que uma imagem é uma matriz, cada execução do algoritmo de detecção de forma vai ser alimentado por uma parte da matriz indicada pelo nó da *max-tree*. Os dados da matriz original não são alterados, pois uma matriz resultante é gerada. Portanto, a detecção de formas executada concorrentemente por regiões não está propensa a condições de disputa [Tanenbaum 2003].

Com isso, a arquitetura do algoritmo de detecção de formas distribuído baseou-se

na distribuição da etapa de detecção de formas em uma plataforma distribuída. Os principais problemas enfrentados têm relação com: identificação do modelo de comunicação mais adequado, escalonamento das regiões (nós da *max-tree*), e otimização do tráfego de mensagens.

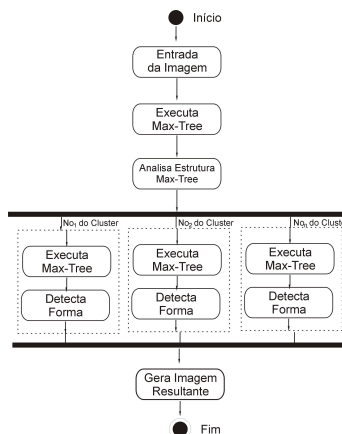


Figura 3. Diagrama de atividades

O diagrama de atividades (Figura 3) representa o relacionamento do mestre com os escravos identificando as atividades de cada um. As atividades *Entrada da Imagem*, *Executa max-tree*, *Analisa estrutura max-tree* e *Gera imagem resultante* são atividades do mestre. Observe que o modelo utilizado trabalha com o princípio de *fork/join* fazendo com que o mestre fique no aguardo (sincronização obrigatória) dos seus escravos para finalizar sua execução com a geração da imagem resultante. A atividade *Analisa estrutura max-tree* é efetuada pelo escalonador, função adicional acrescentada à aplicação distribuída. Aos escravos, dispostos um em cada nó do *Cluster*, cabe as atividades *Executa max-tree* e *Detecta forma*. A definição das atividades aos escravos é responsabilidade do algoritmo de escalonamento [Vieira 2008].

O algoritmo de escalonamento, por si só, é um tema de trabalho. O escalonador levou em consideração, inicialmente, um único aspecto, o número de nós do *Cluster*. Esta decisão de escalonamento foi tomada pela sua simplicidade de implementação. Portanto, a função do escalonador é (1) analisar a imagem de entrada, (2) dividir a imagem em N partes, onde N é o número de nós escravos disponíveis e (3) enviar a imagem a cada escravo.

O terceiro e último problema tratado faz referência à redução (otimização) no número de mensagens trafegadas entre o mestre e os escravos. A solução proposta foi fazer o envio ao escravo somente da porção da imagem que este, efetivamente, vai precisar analisar.

4. Resultados

Com o intuito de validar as premissas estudadas neste projeto, decidiu-se implementar um protótipo do algoritmo de detecção de formas distribuído e avaliar o tempo total de detecção de formas. Os elementos levados em consideração (descritos a seguir) para a obtenção destes resultados foram: (1) protótipo, (2) plataforma, (3) ambiente de teste e (4) cenário.

Programação C com o auxílio da biblioteca de paralelização MPI (*Message Passing Interface*). A versão atual possui um módulo escalonador simplificado que distribui as regiões de forma igualitária independentemente do poder computacional do nó escravo. A plataforma escolhida foi o *Cluster* heterogêneo de estações de Bordeaux/França, pertencente ao Projeto Europeu GRID5000, a qual é composta por dois tipos de estações: *Dell PowerEdge 1855* e *IBM System x3455* (detalhados na Tabela 1). As estações estão interligadas por *Switch Cisco 3750*. Durante a execução, o ambiente de teste encontrava-se “limpo”, isto é, um único usuário estava logado na máquina e esta só possuía os serviços essenciais para execução, entre eles, resalta-se o *dhclient3*, *portmap* e o *rpc.statd*. O cenário definido consiste na detecção da forma de interesse círculos. A imagem utilizada foi uma cartela de aspirina em tons de cinza com resolução de 780x1290, totalizando 1006200 *pixels*. Aplicou-se o algoritmo de detecção de formas seqüencial (original) e distribuído, variando-se, neste último algoritmo, o número de escravos operários.

Tabela 1. Descrição de hardware e software da plataforma de teste

Caraterística	Dell PowerEdge 1855	IBM System x3455
CPU	Intel Xeon EM64T 3GHz	AMD Opteron 2218 2.6 GHz
Memória principal	2GB (4x512MB)	4GB (4x512MB + 2x1024MB)
Memória Cache (L2)	1MB	2MB
Rede	Gigabit ethernet	Gigabit ethernet
Disco	70GB / SCSI	80GB (SATAHitachi)
Sistema Operacional	GNU/Linux Debian 4.2.2-6	GNU/Linux Debian 4.2.2-6
Kernel	2.6.24.3	2.6.24.3

Os testes aplicados consistiam em detectar a forma de interesse círculos na imagem cartela de aspirina. Executou-se o algoritmo seqüencial em uma única estação de trabalho, e posteriormente, o algoritmo distribuído em 3, 7, 11, 15, 19, 23, 27 e 31 estações. O algoritmo distribuído destinava uma estação ao mestre e o número par restante de estações aos escravos. A imagem foi dividida em X partes iguais na altura e Y partes iguais na largura, sendo que no primeiro teste o X foi 2 com Y igual ao divisão do número de escravos por dois, e no segundo teste o X foi 1 com Y igual a número de escravos. Esta divisão já aplica-se ao algoritmo seqüencial obviamente, o que explica a existência de um único ponto no gráfico quando o número de processadores é igual a 1 (seqüencial).

Os resultados obtidos e representados no gráfico (Figura 4) advêm da média de execuções. A amostra considerada foi de 10 execuções para cada ponto representado. Destaca-se que o desvio padrão entre os elementos da amostra é desprezível. A análise deste trabalho baseou-se sobre os fatores de redução do tempo total de detecção. Em seqüencial, a detecção foi efetuada em $\sim 3000s$, distribuído com 3 estações (2 escravos), o tempo obtido chegou a $\sim 519s$, ou seja, uma redução de 83%. Com 11 estações, o tempo de detecção esteve próximo a 1% do tempo seqüencial, estabilizando-se a partir deste ponto. Os demais pontos do gráfico mostram que o tempo de comunicação (transmissão da imagem) passou a ser importante e a carga computacional destinada aos escravos ínfima, não justificando assim a distribuição desta imagem em um número tão elevado de nós. Estes resultados motivam a investigação sobre mecanismos que permitam o processamento de imagens maiores tendo em vista que o poder computacional, neste caso, não foi esgotado. □

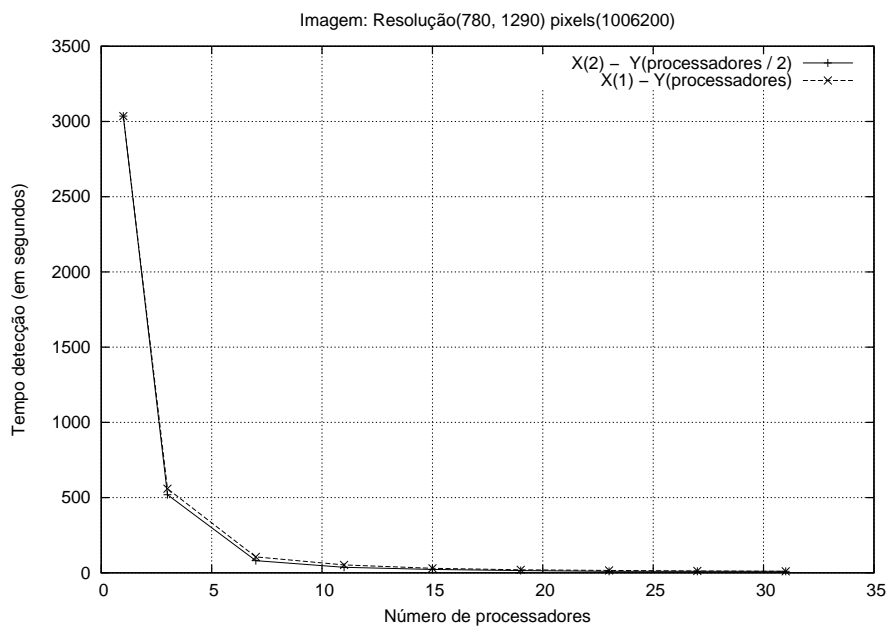


Figura 4. Tempo de detecção vs. Número de processadores

5. Conclusão

A união de técnicas de áreas distintas da ciência em aplicações computacionais não é uma novidade. Neste trabalho, alguns exemplos foram apresentados ligando processamento de imagens e sistemas distribuídos. Contudo, algumas contribuições na área ainda são possíveis, como pode ser demonstrado por este trabalho. A adequação do algoritmo de detecção de formas de interesse (aplicação de processamento de imagens) com a inclusão do modelo de comunicação mestre/escravo (sistemas distribuídos), teoricamente, pode proporcionar duas melhorias à aplicação original: capacidade de processamento de imagens maiores (com maior qualidade) e redução do tempo de detecção.

O foco deste artigo é demonstrar os resultados preliminares do protótipo do algoritmo de detecção de formas distribuído com o objetivo específico de reduzir o tempo de detecção de formas comparando-o com a versão em seqüencial. A versão distribuída do algoritmo está operacional e teve suas execuções lançadas em uma plataforma distribuída de *Clusters* de estações. Os resultados preliminares são motivadores, pois com um incremento de apenas 2 estações (totalizando 1 mestre e 2 escravos comparado com a aplicação original que necessita de 1 estação) obteve-se ganhos próximos a 17%. Em valores absolutos, a aplicação original (1 estação) finalizou as suas tarefas com tempos superiores a 3000s e a aplicação distribuída com tempos inferiores a 10s (com mais de 20 estações).

O conhecimento adquirido com o estudo nas áreas de processamento de imagens e sistemas distribuídos bem como os resultados preliminares apresentados neste trabalho permitem identificar alguns pontos que ainda podem ser explorados. Os resultados obtidos comprovaram a premissa inicial deste trabalho sobre a redução de tempo de detecção de formas para o algoritmo distribuído e motivam a investigação de mecanismos que permitam o aumento da capacidade de processamento de imagens com dimensões maiores. Espera-se, ainda, alterar a forma de divisão da imagem facilitando as decisões do escalonador na distribuição das tarefas. Além disso, um novo escalonador deverá ponderar elementos como: potência de cálculo de cada escravo, número de *pixels* e número de regiões que compreende a porção da imagem destinada a cada escravo. □

Acknowledgment

Experiments presented in this paper were carried out using the Grid'5000 experimental testbed, an initiative from the French Ministry of Research through the ACI GRID incentive action, INRIA, CNRS and RENATER and other contributing partners (see <https://www.grid5000.fr>).

Referências

- Cassol, G. L. F. (2005). Detecção de formas usando representação hierárquica de regiões para imagens em níveis de cinza. Trabalho de diplomação (bacharelado em ciência da computação), Departamento de Ciência da Computação, UDESC/CCT, Joinville.
- Coulouris, G., Dollimore, J., and Kindberg, T. (2005). *Distributed Systems Concepts and Design*. Addison Wesley, 5nd edition.
- Facon, J. (2002). Processamento de imagens. Master's thesis, Pontifícia Universidade Católica do Paraná.
- Fink, G. A., Jungclaus, N., Kummert, F., Ftitter, H., and Sagerer, G. (1996). A distributed system for integrated speech and image understanding. *IEEE TRANSACTIONS ON SYSTEMS*, pages 117–126.
- Nicolescu, C. and Jonker, P. (2000). Parallel low-level image processing on a distributed-memory system. In *IPDPS '00: Proceedings of the 15 IPDPS 2000 Workshops on Parallel and Distributed Processing*, pages 226–233, London, UK. Springer-Verlag.
- Ostermann, L. G. (2002). *Hierarchical Region Based Processing of Images and Video Sequences: Application to Filtering, Segmentation and Information Retrieval*. PhD thesis, Department of Signal Theory and Communications - Universitat Politècnica de Catalunya.
- Salembier, P., Oliveras, A., and Garrido, L. (1998). Antiextensive Connected Operators for Image and Sequence Processing. *IEEE Transactions on Image Processing*, 7(4):555–570.
- Silva, A. G. and de Alencar Lotufo, R. (2007). Hierarchical morphological analysis for generic detection of shapes in grayscale images. *Computational Modelling of Objects Represented in Images*, page 405 a 410.
- Silva, A. G., de Alencar Lotufo, R., Felipussi, S. C., and Cassol, G. L. F. (2006). K-max - segmentatin based on selection of max-tree deep nodes. *Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning*.
- Subramanya, S. R. (1999). A distributed algorithm for the classification of images on a network of workstations. *ACM international conference on Multimedia*.
- Tanenbaum, A. S. (2003). *Sistemas Operacionais Modernos*. Pearson Prentice Hall, 2nd edição edition.
- Vieira, C. (2008). Distribuição de um algoritmo de detecção de formas. Trabalho de diplomação (bacharelado em ciência da computação), Departamento de Ciência da Computação, UDESC/CCT, Joinville.

