Segmentação de Displasias Corticais Focais em Imagens de Ressonância Magnética do Cérebro Humano

Este exemplar corresponde à redação final da Tese devidamente corrigida e defendida por Felipe P.G. Bergo e aprovada pela Banca Examinadora.

Campinas, 2 de Abril de 2008.

Prof. Dr. Alexandre X. Falcão (Orientador)

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

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Bibliotecária: Maria Júlia Milani Rodrigues – CRB8a / 2116

Bergo, Felipe Paulo Guazzi

B454s Segmentação de displasias corticais focais em imagens de ressonância magnética do cérebro humano / Felipe Paulo Guazzi Bergo -- Campinas, [S.P. :s.n.], 2008.

Orientador : Alexandre Xavier Falcão

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

1. Processamento de imagens. 2. Ressonância magnética. 3. Neurociência. 4. Epilepsia. I. Falcão, Alexandre Xavier. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Título em inglês: Focal cortical displasia segmentation in magnetic resonance images of the human brain.

Palavras-chave em inglês (Keywords): 1. Image processing. 2. Magnetic resonance. 3. Neuroscience. 4. Epilepsy.

Área de concentração: Ciência da Computação

Titulação: Doutor em Ciência da Computação

Banca examinadora:

Prof. Dr. Alexandre Xavier Falcão (IC-UNICAMP)
Prof. Dr. Sergio Shiguemi Furuie (INCOR)
Profa. Dra. Agma Juci Machado Traina (ICMC-USP)
Prof. Dr. Fernando Cendes (FCM-UNICAMP)
Prof. Dr. Roberto de Alencar Lotufo (FEEC-UNICAMP)

Data da defesa: 02/04/2008

Programa de Pós-Graduação: Doutorado em Ciência da Computação

TERMO DE APROVAÇÃO

Tese Defendida e Aprovada em 02 de abril de 2008, pela Banca examinadora composta pelos Professores Doutores:

Profª. Drª. Agma Juci Machado Traina
ICMC/DCE – USP.

Prof. Dr. Sérgio Shiguemi Furuie
InCor – USP.

Prof. Dr. Fernando Cendes
FCM – UNICAMP.

Prof. Dr. Roberto de Alencar Lotufo
FEEC – UNICAMP.

Prof. Dr. Alexandre Xavier Falcão
IC – UNICAMP.

Instituto de Computação
Universidade Estadual de Campinas

Segmentação de Displasias Corticais Focais em Imagens de Ressonância Magnética do Cérebro Humano

Felipe P.G. Bergo[1]

Abril de 2008

Banca Examinadora:

- Prof. Dr. Alexandre X. Falcão (Orientador)
- Prof. Dr. Sergio Shiguemi Furuie
  Instituto do Coração (HC/FMUSP)
- Prof^a. Dra. Agma Juci Machado Traina
  ICMC – Universidade de São Paulo
- Prof. Dr. Fernando Cendes
  FCM – Universidade Estadual de Campinas
- Prof. Dr. Roberto de Alencar Lotufo
  FEEC – Universidade Estadual de Campinas
- Prof. Dr. Siome Klein Goldenstein (Suplente)
  IC - Universidade Estadual de Campinas
- Prof. Dr. Nelson Delfino d'Ávila Mascarenhas (Suplente)
  DC – Universidade Federal de São Carlos

[1]Suporte financeiro de: Bolsa CAPES 2004–2007, Bolsa PED A Unicamp 2007–2008.

Resumo

O diagnóstico médico por imagem é uma tarefa complexa, que depende da avaliação subjetiva de um grande volume de dados. Diversas possibilidades de patologia não são consideradas por limitação de tempo e especialização dos profissionais da área médica, mesmo quando os exames adequados estão disponíveis. O desenvolvimento de técnicas automáticas de auxílio ao diagnóstico é um avanço importante para simplificar a tarefa do médico, descartando um grande número de patologias ou indicando as localizações mais prováveis de eventuais anormalidades patológicas.

Displasias corticais focais (FCDs) estão associadas à epilepsia, e são uma das causas mais comuns de casos de epilepsia refratária, em que o tratamento medicamentoso não é suficiente para controlar as crises. As FCDs são lesões que geram variações locais e sutis na aparência do tecido em imagens de ressonância magnética (RM). Seu diagnóstico é em geral uma tarefa difícil e subjetiva. Detecção e localização de eventuais lesões de FCD são passos fundamentais para o planejamento do tratamento do paciente.

Neste trabalho propomos um método para segmentação automática de FCDs em imagens de ressonância magnética (RM) tri-dimensional do cérebro humano. Desenvolvemos novas técnicas de segmentação e análise de imagens, automatizamos uma técnica previamente interativa (reformatação curvilinear) e, através de classificação por aprendizado supervisionado, obtivemos detecção de 100% das lesões, com cobertura de 76,9% dos voxels lesionais. Este resultado é um pouco melhor que o estado da arte, embora ainda não seja uma solução ideal, solidamente validada, para o problema.

Abstract

Medical diagnosis from imaging techniques is a complex task that depends on subjective evaluation of a large volume of data. Many pathologies are often not considered due to time and experience restrictions of the medical crew, even when the imaging data are readily available. The development of computer-aided diagnosis techniques greatly simplify the physician's work, by discarding a large number of pathologies and/or pointing out the most probable locations of pathological abnormalities.

Focal cortical displasia (FCDs) are associated to epilepsy, and are one of the most common causes of refractory epilepsy, where drug-based treatment does not eliminate the seizures. FCDs are lesions that lead to subtle, localized appearance variations of brain tissue in magnetic resonance (MR) imaging. Their diagnosis is difficult, tedious and subjective. Detection and localiation of FCD lesions are key steps for treatment planning.

In this work we propose a method for automatic segmentation of FCDs in tridimensional magnetic MR images of the human brain. We developed new image segmentation and image analysis techniques, automated a previously interactive technique (curvilinear reformatting) and, through classification by supervised learning, achieved detection of 100% of the lesions, with 76,9% coverage of the lesional voxels. This result is slightly better than the state-of-the-art, even though it still is has not been thoroughly validated on a large data base and can still be improved.

Agradecimentos

A todos que contribuiram diretamento no desenvolvimento deste trabalho: Alexandre Falcão (orientador) e o corpo docente do Instituto de Computação da Unicamp; Prof. Dr. Fernando Cendes, Dra. Clarissa Yasuda e todos os pesquisadores do Laboratório de Neuro-Imagem do Hospital das Clinicas da Unicamp.

A todos os colegas do Laboratório de Informática Visual do Instituto de Computação da Unicamp, criado durante a realização deste trabalho e fundamental para sua conclusão.

À minha família por todo o apoio e motivação.

À Capes e à Unicamp pelo apoio financeiro.

Sumário

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

1.1 Motivação

O diagnóstico médico por imagem evoluiu muito nas últimas décadas. Resolução, precisão e protocolos se aperfeiçoaram, e a comunidade médica passou a utilizar e confiar cada vez mais nas técnicas modernas de exame por imagem – ressonância magnética (RM), tomografia computadorizada de raios X, PET e SPECT.

As técnicas de imageamento, por sua vez, oferecem um volume de dados cada vez maior (imagens volumétricas com mais de 100 fatias, por exemplo), o que torna o diagnóstico por inspeção visual um processo lento, tedioso e subjetivo. Com isso surgiu a demanda por métodos de assistência ao diagnóstico, que apontem para o especialista as regiões mais prováveis de anormalidades/patologias, ou garantam com segurança a ausência de tipos específicos de patologias.

Displasias corticais focais (FCD) são mal-formações ocorridas no desenvolvimento do córtex cerebral que resultam em posicionamento anômalo de elementos gliais e falhas na composição laminar do córtex. FCDs foram descritas por Taylor em 1971 [90], e são as mal-formações mais comuns em casos de epilepsia refratária – casos em que o paciente não reage ao tratamento medicamentoso [48]. FCDs são uma causa particularmente comum de epilepsia em crianças e adolescentes [24, 81]. A detecção e localização das FCDs são passos cruciais para o planejamento do tratamento dos pacientes, que em geral envolve a remoção cirúrgica das lesões [75, 84, 64].

FCDs são lesões microscópicas no tecido cortical, mas que geram artefatos macroscópicos sutis em imagens de ressonância magnética: borramento da transição entre substância branca e substância cinzenta, sinal T1 hiperintenso na substância cinzenta e espessamentos localizados do córtex [24]. Formas especiais de visualização de imagens de RM facilitam o diagnóstico [7, 72], mas a tarefa permanece sendo tediosa e subjetiva.

A detecção/segmentação automatizada de FCDs em imagens de RM volumétrica T1 tem sido objeto de diversos trabalhos nos últimos anos [60, 2, 86, 23, 22]. Entretanto, a maioria dos métodos se baseia em métodos de segmentação do cérebro e de WM/GM (substância branca/cinzenta) que dependem de modelos (templates), e não são precisos ou adequados a imagens de crianças ou pacientes operados. A segmentação de WM/GM em particular é uma tarefa complicada, já que dois dos artefatos causados pelas FCDs são o borramento da transição WM/GM e intensidades anômalas na substância cinzenta lesionada.

1.2 Visão Geral das Contribuições

A principal contribuição desta tese é um método de segmentação de displasias corticais focais, aplicável a imagens de pacientes de qualquer idade, e também a pacientes operados. Além do método de segmentação em si, este trabalho também contribuiu com uma base de 20 lesões de displasia delineadas, útil para a avaliação e melhoria de métodos de segmentação de displasias. Outra contribuição importante é um algoritmo particionado para a IFT, que permite um melhor aproveitamento de computadores com múltiplos núcleos de processamento (CPUs) para tarefas de processamento de imagens.

Para desenvolver o método de segmentação de displasias, utilizamos em todos os passos técnicas que independem de modelos (templates). Evitamos a utilização de técnicas de segmentação de substância branca/cinzenta, por não termos uma avaliação adequada da confiabilidade deste tipo de segmentação em pacientes com FCDs. O principal diferencial do método desenvolvido é utilizar medidas de assimetria de textura, imitando o procedimento de diagnóstico visual: a diferença de borramento e/ou intensidade entre regiões simétricas é um indicador mais forte de presença de lesão do que medidas absolutas realizadas apenas ao redor do local avaliado. A Figura 1.1 mostra um diagrama dos 5 passos envolvidos no método.

Segmentação Automática do Cérebro. O primeiro passo do método é a segmentação automática do cérebro, ilustrada na Figura 1.2. Para realizar este passo, desenvolvemos uma nova técnica de segmentação baseada em grafos, a segmentação por poda de árvores [37, 69, 12]. Esta técnica, apresentada em detalhes no Capítulo 2, oferece segmentação robusta e eficiente mesmo em imagens ruidosas como as obtidas nos exames de ressonância magnética.

Localização do Plano Inter-Hemisférico. O segundo passo é a localização do plano inter-hemisférico (*mid-sagittal plane*, nas publicações em inglês), ilustrada na Figura 1.3. Este plano delimita os hemisférios cerebrais esquerdo e direito, e sua localização é fundamental para extrair as medidas de assimetria de textura nas do passo 4. Desenvolvemos

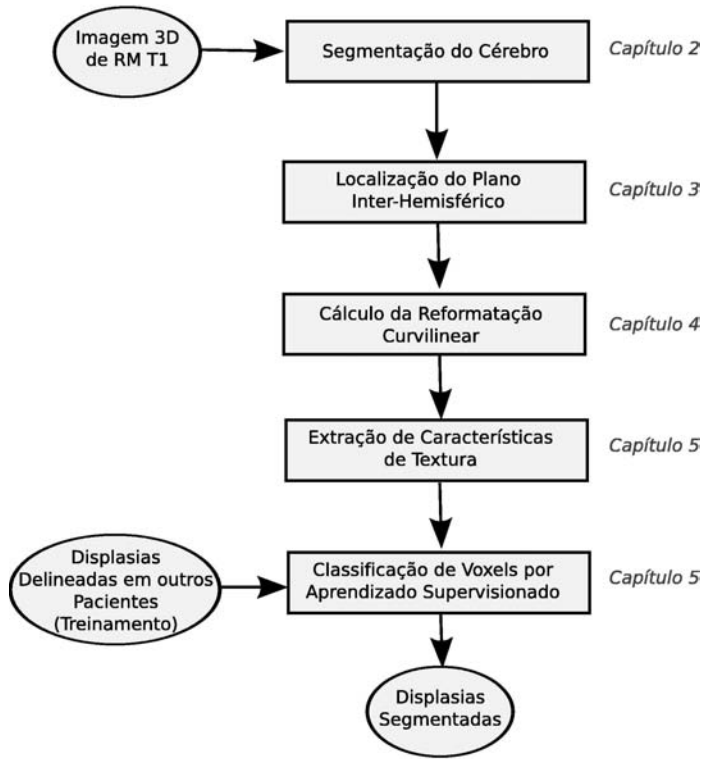
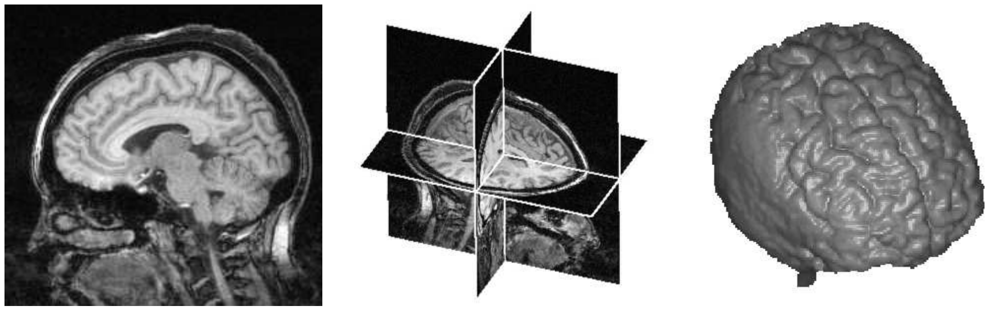


Figura 1.1: Diagrama do método de segmentação de displasias corticais focais.

um método de localização do plano inter-hemisférico [14] mais robusto que o estado da arte [98]. Este método é discutido em detalhes no Capítulo 3.

Cálculo da Reformatação Curvilinear. O terceiro passo é o cálculo da reformatação curvilinear [7], ilustrada na Figura 1.4. A reformatação curvilinear consiste em visualizar as intensidades do volume de RM como superfícies eqüidistantes da superfície externa do córtex cerebral. A implementação original desta técnica, em um software comercial [79], exigia interação considerável do usuário para calcular a reformatação curvilinear. Desenvolvemos uma técnica para obter a reformatação curvilinear de forma totalmente automática [10]. Esta técnica foi utilizada com sucesso para auxiliar o diagnóstico visual [72], e o nosso método automático de segmentação de displasias tenta replicar este procedimento de diagnóstico, comparando assimetrias de textura sobre a reformatação curvilinear. Esta contribuição é discutida em detalhes no Capítulo 4.

Extração de Características de Textura. No quarto passo, extraímos de cada voxel do cérebro um par de texturas simétricas [13], como ilustrado na Figura 1.5(a). Realizamos



(a) (b) (c)

Figura 1.2: Segmentação automática do cérebro: (a) Exemplo de imagem de ressonância magnética. (b) Um volume típico de RM do cérebro é formado por aproximadamente 120–200 imagens como a mostrada em (a). (c) Renderização 3D do cérebro segmentado automaticamente por poda de árvores a partir do volume em (a)–(b).

experimentos com diversas dimensões de textura, variando de 8×8 mm^2 a 64×64 mm^2, e obtivemos melhores resultados com texturas de 16×16 mm^2 (nas imagens de nossos experimentos, 1 pixel = 1 mm^2). Exemplos de pares de texturas (uma sobre uma lesão de FCD, a outra sobre tecido normal em posição simétrica) são ilustrados na Figura 1.5(b). Após a extração das texturas, computamos diversas medidas quantitativas de borramento, intensidade, contraste e homogeneidade, que formam um vetor de características associado a um voxel do volume de RM.

Classificação de Voxels por Aprendizado Supervisionado. O quinto e último passo é a classificação de voxels por aprendizado supervisionado [36], onde características computadas em exemplos de segmentação são usadas para “ensinar” um programa de computador a diferenciar classes de objetos. No nosso caso, a tarefa é diferenciar voxels sadios de voxels pertencentes a FCDs, utilizando características da textura tangente à reformatação curvilinear [7, 10, 13]. As delineações de lesões para realizar o aprendizado foram realizadas pela Dr. Clarissa Yasuda (Departamento de Neurologia, FCM, Unicamp) e supervisionadas pelo Prof. Dr. Fernando Cendes (Departamento de Neurologia, FCM, Unicamp). As lesões de 20 pacientes foram delineadas e utilizadas para o deenvolvimento do método. Detalhes sobre o software desenvolvido para delineação das lesões são apresentados no Apêndice A.

O método proposto foi avaliado em 5 lesões de FCD frontal e obteve detecção de 100% das lesões, com uma taxa de cobertura de 76,9% (voxels lesionais corretamente identificados), valor superior ao do estado da arte (73% [23]). Considerações mais detalhadas sobre o método proposto são feitas no Capítulo 7.

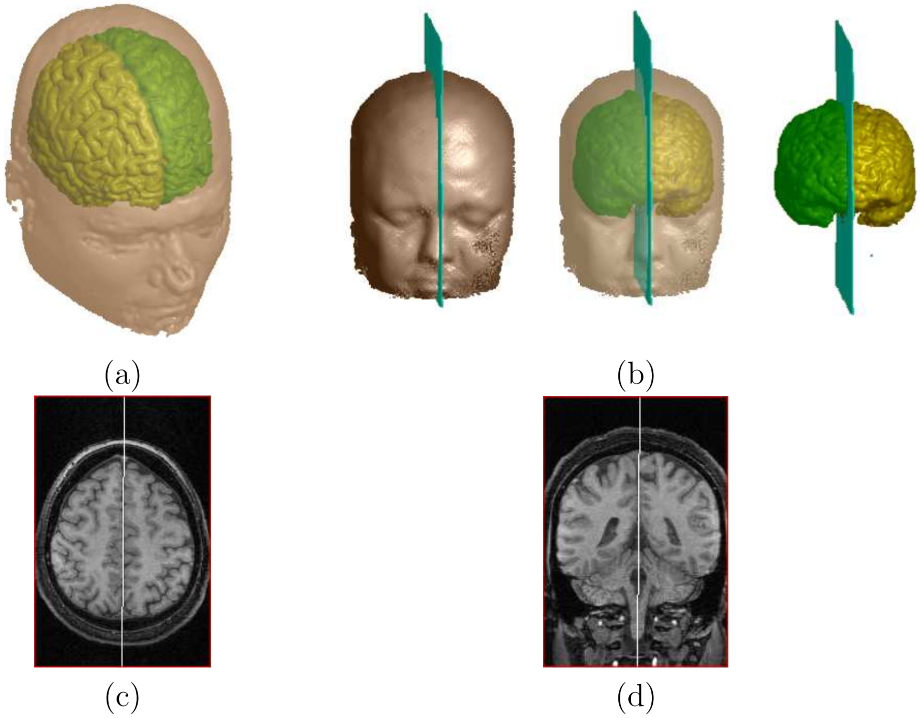


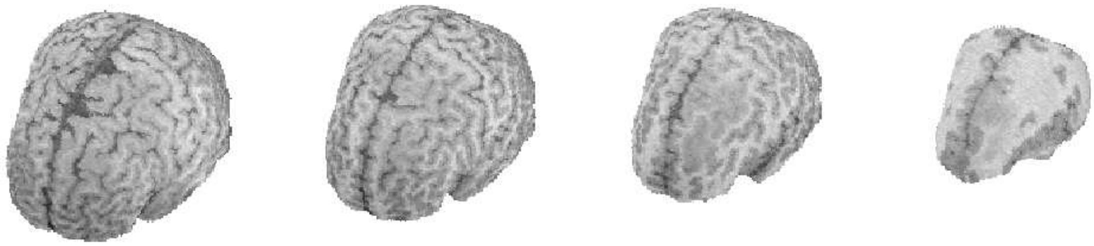
Figura 1.3: Hemisférios cerebrais e plano inter-hemisférico: (a) hemisférios esquerdo (verde) e direito (amarelo) em renderização 3D de um volume de ressonância magnética. (b) Plano inter-hemisférico (ciano) em renderizações 3D de um volume de RM e (c–d) Plano inter-hemisférico traçado sobre fatias 2D do volume de RM.

1.3 Organização da Tese

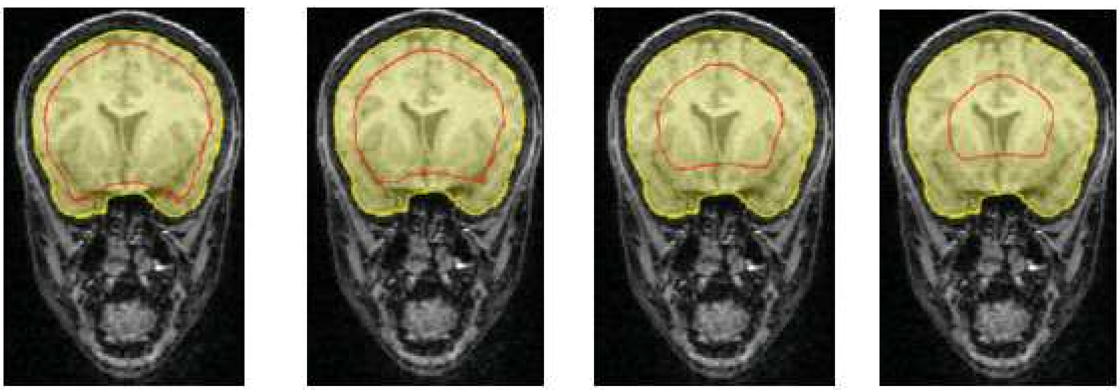
O texto desta tese está organizado agrupando os principais artigos publicados ou aceitos para publicação como resultado da pesquisa realizada.

1.3.1 Capítulo 2

Este capítulo (Automatic Image Segmentation by Tree Pruning) inclui os resultados publicados em [12]. A técnica de segmentação por poda de árvores (tree pruning) foi idealizada como uma forma de explorar as propriedades da floresta de caminhos mínimos obtida na implementação da transformada de Watershed [96, 15] através da Transformada Imagem-Floresta (IFT) [39]. A poda de árvores explora propriedades da dinâmica de propagação ordenada. As bordas de objetos em imagens não-sintéticas em geral não são homogêneas, e possuem trechos onde a fronteira entre objeto e fundo é mais sutil.



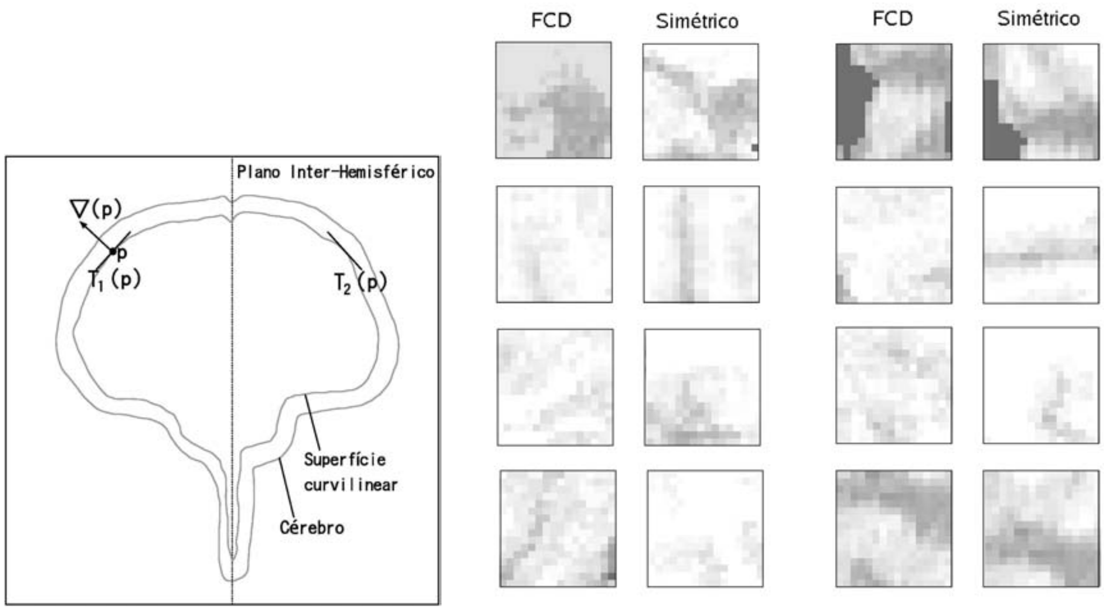
(a)



(b)

Figura 1.4: Reformatação curvilinear: (a) Exemplos da reformatação curvilinear de um volume de RM, exibindo superfícies em 5 profundidades distintas. (b) Exemplos de superfícies eqüidistantes à superfície externa do córtex, traçadas (em vermelho) sobre uma fatia 2D da RM.

Algoritmos de crescimento ótimo de regiões [78, 95, 39, 40] tendem a "vazar" do interior do objeto para o fundo por esses trechos de fronteira mais fraca. Esta propriedade forma nós de vazamento na floresta de caminhos mínimos da IFT que podem ser identificados de forma automática. A poda das árvores nos nós de vazamento permite a segmentação de imagens por crescimento de regiões sem a necessidade de sementes no fundo da imagem. Esta vantagem é particularmente importante em aplicações onde não há controle sobre as propriedades do fundo, ou este contém um grande número de falsas bordas que tornariam a segmentação ambígua. A primeira formulação e validação da poda de árvores foi publicada no SIBGRAPI de 2004 [37], com uma implementação semi-interativa. A ferramenta implementada exibia um grande número de nós candidatos para poda e o usuário selecionava interativamente um conjunto satisfatório de pontos de poda. O algo-



(a) (b)

Figura 1.5: Extração de características de textura: (a) Texturas simétricas T_1 e T_2. A simetria é obtida pelo plano inter-hemisférico, e a orientação da textura – $\nabla(p)$ – é dada pela reformatação curvilinear. (b) Exemplos de pares de texturas 16×16 mm^2.

ritmo de detecção de pontos de poda foi aperfeiçoado e totalmente automatizado. Esse resultado foi apresentado no SIBGRAPI de 2006 [69]. Os resultados destes dois trabalhos foram consolidados e apresentados com maiores detalhes em artigo publicado no Journal of Mathematical Imaging and Vision em 2007 [12], apresentado neste capítulo.

1.3.2 Capítulo 3

Este capítulo (Fast and Robust Mid-sagittal Plane Location in 3D MR Images of the Brain) inclui os resultados publicados em [14]. A detecção do plano de simetria entre os hemisférios é um passo crucial para a técnica de segmentação de displasias, já que o método proposto se baseia na comparação de texturas simetricamente opostas. O estado da arte [98] apresenta técnicas extremamente rápidas (menos que 10 segundos por imagem volumétrica de RM do cérebro), porém com muitos parâmetros arbitrários e sem avaliação confiável em cérebros com patologias ou com variações estruturais (tais como crianças e idosos). Desenvolvemos nosso próprio método, utilizando a segmentação do cérebro por poda de árvores [12] e operações morfológicas [35] para excluir lesões do espaço de busca, seguido por uma heurística de busca com passos de ajuste fino e grosso que encontram o plano de simetria com precisão bem próxima ao limite teórico, sem ficar presa a falsos máximos locais ou artefatos de lesões que poderiam se confundir com a fenda inter-hemisférica. O artigo [14] avaliou o método em 64 volumes, incluindo controles (20), pacientes com lesões (6), pacientes operados (30) e imagens sintéticas (8). Nosso método leva em torno de 60 segundos por volume e obteve sucesso em todas os volumes testados, com variação angular de 1,26 grau, valor bem próximo ao limite teórico de 0,5 grau, e de difícil percepção visual. Após esta publicação, o método foi aperfeiçoado para ser mais rápido, realizando a etapa de segmentação em resolução reduzida, sem perda de precisão na localização do plano de simetria. A avaliação dos efeitos da redução de resolução na segmentação para a localização do plano de simetria e comparação com o método de Volkau [98] na mesma base de imagens está em andamento, e será submetida como artigo de revista em breve.

1.3.3 Capítulo 4

Este capítulo (Fast and Automatic Curvilinear Reformatting of MR Images of the Brain for Diagnosis of Dysplastic Lesions) inclui os resultados publicados em [10]. Este trabalho apresenta um método rápido e automático para computar a reformatação curvilinear do cérebro. A reformatação curvilinear foi apresentada por Bastos et. al [7] e mostrou-se uma ferramenta muito útil para aumentar a precisão no diagnóstico de FCDs [72]. A reformatação curvilinear consiste em visualizar as intensidades da ressonância magnética em superfícies eqüidistantes da superfície externa do cérebro. Esta forma de visualização evita que artefatos de curvatura do córtex e volumes parciais sejam confundidos com lesões displásticas. Como a premissa do trabalho desta tese é analisar automaticamente a reformatação curvilinear, sua computação é um requisito primordial. As iso-superfícies do cérebro podem ser codificadas como uma transformada de distância Euclideana (TDE) a partir da superfície externa do córtex, e a visualização de uma profundidade específica pode ser realizada por projeção de voxels em um intervalo limitado de distâncias. A segmentação do cérebro é realizada com a poda de árvores automática (Capítulo 2 [12]) e a TDE é computada de forma eficiente através de uma IFT [39].

1.3.4 Capítulo 5

Este capítulo (FCD Segmentation using Texture Asymmetry of MR-T1 Images of the Brain) inclui os resultados aceitos para publicação em [13]. Este trabalho apresenta um método automático para segmentação de lesões displásticas a partir de aprendizado supervisionado. O método realiza segmentação automática do cérebro através da poda de árvores [12], computa a reformatação curvilinear através de uma IFT-TDE [10], encontra o plano inter-hemisférico através de uma busca heurística [14] e utiliza um classificador Reduced Coulomb Energy (RCE) [36] para segmentar displasias corticais focais. O vetor de características utilizado explora assimetrias de textura entre os hemisférios, de forma similar ao procedimento de diagnóstico por inspeção visual [72]. As lesões foram segmentadas manualmente em 20 volumes por uma especialista (Dra. Clarissa L. Yasuda), e esta segmentação foi utilizada como treinamento para o processo de aprendizado do classificador. Neste artigo realizamos experimentos com 5 pacientes (2 deles crianças), por restrição de tempo e recursos computacionais à época do período de submissão. Obtivemos detecção de 100% das lesões, com taxa de cobertura (voxels lesionais corretamente identificados) de 76,9%. Este valor é superior ao estado da arte, que obteve 73% [23]. Além de obter um resultado quantitativamente melhor, nosso método não depende de segmentações baseadas em templates como os demais trabalhos da literatura, sendo aplicável a pacientes de qualquer idade e/ou com outras anomalias estruturais (tais como intervenções cirúrgicas).

1.3.5 Capítulo 6

Este capítulo (A Partitioned Algorithm for the Image Foresting Transform) inclui os resultados publicados em [11]. Neste trabalho apresentamos um algoritmo para particionar a computação de qualquer operador baseado na IFT [39] entre vários nós – multicomputadores e redes de computadores. Este particionamento pode ser utilizado para paralelizar uma IFT, ou para dividir uma IFT em vários passos menores. Esta última solução permite a computação de IFTs em grandes imagens/volumes em dispositivos com memória limitada. Ao contrário dos algoritmos específicos para paralelização de transformadas de watershed [70, 71] e transformadas de distância [18], a IFT particionada [11] é uma solução genérica de paralelização/particionamento, aplicável a qualquer operador baseado na IFT. Embora a IFT particionada não tenha sido utilizada no método de segmentação de displasias, ela é aplicável aos passos de segmentação do cérebro [12] e cálculo da reformatação curvilinear [10].

Capítulo 2

Automatic Image Segmentation by Tree Pruning

2.1 Introduction

The problem of defining the precise spatial extent of a desired object in a given image, namely *image segmentation*, is addressed. The difficulties stem from the absence of global object information (location, shape, appearance) and the similarities between object and background with respect to image features (color and texture). These difficulties usually call for user assistance [59, 26, 25, 44, 16, 40], making automatic segmentation viable only in an application-dependent and tailored fashion. Methods for automatic segmentation should separate the part that is application-dependent from the application-independent part, such that the former can be easily tailored for different applications. We present a method, called *tree pruning*, which is consistent with this strategy.

Tree pruning uses the *Image Foresting Transform* (IFT)— a tool for the design of image processing operators based on connectivity [39]. The IFT has been applied to compute distance transforms, multiscale skeletonizations, morphological reconstructions, watershed transforms, boundary tracking, fractal dimension, and shape saliences [43, 65, 42, 41, 66, 93, 92]. Tree pruning is the first IFT-based operator that exploits a *combinatorial property* of the forest — the number of descendants that each node has in the image's border.

In the IFT framework, an image is interpreted as a graph whose nodes are image pixels and whose arcs are defined by an *adjacency relation* between pixels. For a given set of *seed pixels* and suitable *path-cost function*, the IFT computes an optimum-path forest in the graph whose roots are drawn from the seed set. Each tree in the forest consists of pixels more strongly connected to its root than to any other seed. In tree pruning, the seeds are chosen inside the object and the choice of the path-cost function intends to connect object and background by optimum paths (*leaking paths*), which cross the object's boundary through its "most weakly connected" parts (*leaking pixels*). The above combinatorial property is exploited to automatically identify the leaking pixels and eliminate their subtrees, such that the remaining forest defines the object.

Tree pruning runs in time proportional to the number of pixels, is extensible to multidimensional images, is free of *ad hoc* parameters, and requires only internal seeds, with little interference from the heterogeneity of the background. These aspects favor solutions which exploit image features and object information for automatic segmentation. For example, we can estimate seeds using approaches for object location [97]. Candidate seeds can otherwise be used to obtain a set of possible objects, and the desired one can be chosen based on objective functions [83, 99, 61] or object features and pattern classifiers [53, 36, 62]. One can also exploit some combination between tree pruning and deformable models [59, 26, 25, 20] in order to achieve a better agreement between the geometry of the model and local image features. Even in the context of interactive segmentation, it is highly desirable to make the user's actions simple and minimal. Tree pruning reduces user intervention to a few markers in the image.

In comparison with region-growing approaches based on optimum paths from internal seeds [73, 95], the criterion to disconnect object and background does not use the costs of the optimum paths but the above combinatorial property. Optimum paths that reach object pixels are assumed to not pass through the background, instead of having costs strictly lower than the costs of paths that reach the background. In tree pruning, internal seeds compete among themselves and only a few seeds become roots of leaking paths. Other approaches based on optimal seed competition [65, 15, 96, 82, 80] can be roughly described in three steps: (i) seed pixels are selected inside and outside the objects, (ii) each seed defines an *influence zone* that contains pixels which are more strongly connected to that seed than to any other, and (iii) each object is defined by the union of the influence zones of its internal seeds. The absence of boundary information and/or heterogeneity of the background usually cause invasion (leaking) of object seeds in influence zones of background seeds and vice-versa. In interactive segmentation, the user corrects leaking by adding and removing seeds. In the context of the IFT, these corrections can be done in sublinear time [40, 4] (e.g., by differential watershed transforms). Tree pruning exploits the leaking problem and also favors solutions for automatic segmentation, as discussed above. On the other hand, tree pruning and the IFT-watershed transform by markers [65] use the same image graph and path-cost function which makes them competitive approaches when the external seeds for watershed can be found automatically. This aspect is also evaluated in Section 2.6.

Approaches for image segmentation usually exploit image features and some object information to emphasize the discontinuities between object and background. It is desirable, for example, that the external energy in snakes be lower along the object's boundary than inside and outside it [59]; the arc costs in live wire be lower along the object's boundary than within a neighborhood around it [44]; the local affinities in relative fuzzy connectedness [82] be higher inside and outside the object than on its boundary; the gradient values in watershed transform be higher for pixels on the object's boundary than inside and outside it [65, 15, 96]; and the arc weights in graph-cut segmentation be lower across the object's boundary than inside and outside it [16, 83, 61]. Additionally, the energy minimization in [16, 61] using min-cut/max-flow algorithms from source to sink nodes [46] also requires lower arc-weights between source and object pixels, higher arc-weights between sink and object pixels, lower arc-weights between sink and background pixels, and higher arc-weights between source and background pixels. Clearly the effectiveness of these approaches is affected when the above *desirable conditions* are not fully satisfied, but they can still work under certain hard constraints (usually, involving the user). For example, Boykov and Jolly [16] allow the user to force the arc weights with source and sink by selecting seed pixels inside and outside the object.

Tree pruning can take advantage of the same image features and object information to create a *gradient-like image* (Figure 2.1a) in which pixel values are higher on the object's boundary than inside it and, at least, in a neighborhood outside it. The *cost of a path* in tree pruning is given by the maximum gradient value along it. Considering all possible paths from the internal seed set to a given pixel, the IFT assigns a path of minimum cost to that pixel. Therefore, the leaking pixels will be those with lower values along the object's boundary and the leaking paths will reach all background pixels around the object with costs equal to their leaking pixel values. By connectivity, the rest of the background will be also conquered by leaking paths that pass through the same leaking pixels. As explained above, the automatic identification of these leaking pixels solves the problem (Figure 2.1b). Under the same conditions, the watershed transform may fail whenever the heterogeneity of the background results in gradient values similar to those of the desired boundary, because the costs of optimum paths from external seeds may saturate before these paths reach the internal ones at the object's boundary (Figure 2.1c). This will make the location of the external seeds more important to solve the problem in watershed-based approaches (see arrow in Figure 2.1d).

Tree pruning was first presented in [37], with two approaches to detect leaking pixels. One is interactive where the user can visually identify leaking pixels and select them with the mouse pointer. The other is automatic, but relies on a parameter that is difficult to be adjusted in real applications. In [69], we revisited the method to propose an automatic solution for leaking pixel detection, which is free of *ad hoc* parameters, and to provide a comparative analysis of tree pruning and watershed transform for automatic segmentation, including one experiment for license plate segmentation. In this paper, we provide

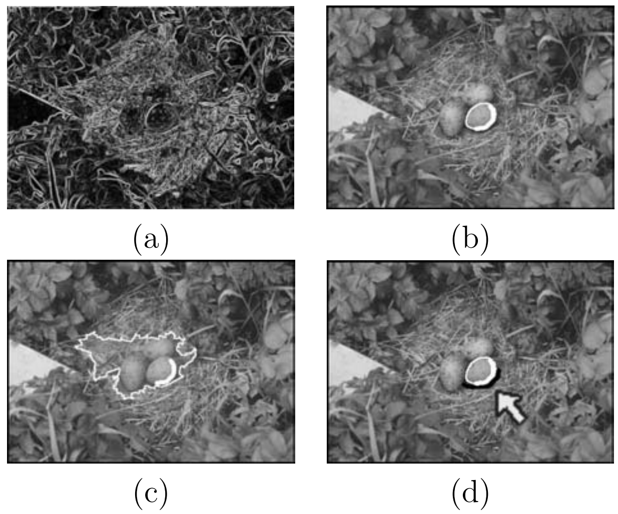


Figura 2.1: (a) A gradient-like image where the object is an egg. (b) Segmentation result with tree pruning and the white marker as internal seeds. (c–d) Segmentation results with watershed under the same conditions and using the image's border and the black marker (indicated by the arrow) as external seeds.

more details in the presentation of the method (formal definition of the obtained objects, different examples, algorithms, sufficient conditions, gradient-like images, and geometrical issues), improve the license plate segmentation method and presentation, and add another application (automatic 3D MR-brain segmentation), in which the methods are compared with a template-based approach widely used for medical research [47].

We give the main definitions and instantiate the IFT for tree pruning and watershed transform in Section 2.2; describe tree pruning with algorithms in Section 2.3; discuss sufficient conditions and geometrical issues in Section 2.4 and gradient-like images in Section 2.5; evaluate the methods in Section 2.6; and state conclusions in Section 2.7.

2.2 Background

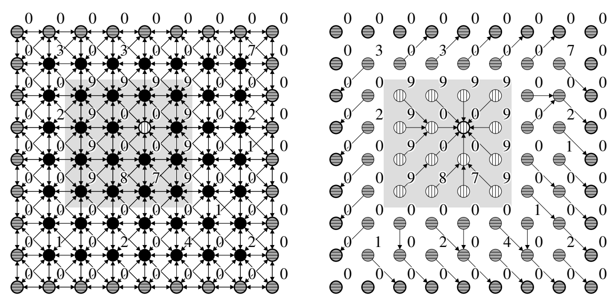
Tree pruning (TP) and watershed (WS) algorithms rely on a *gradient-like image* (gradient image for short), being both approaches extensible to multidimensional and multiparametric images. In several situations, the gradient image can be simply the magnitude of some gradient operator, such as the Sobel's gradient (Figure 2.1a). In other situations, it is better to assign an image feature vector to each pixel and compute the gradient image as a function of the differences between feature vectors of adjacent pixels (Section 2.5).

Gradient condition: In WS, it is desirable to have a gradient image with higher pixel values on the object's boundary than inside and outside the object. In TP, the lower pixel values outside the object are desirable only within a small neighborhood around the object's boundary. These gradient conditions are important to understand the methods, but they are not necessary conditions (see Section 2.4).

In the following, we present the image foresting transform and its algorithm for TP and WS.

2.2.1 Image Foresting Transform

A gradient image $\hat{I}$ is a pair (D_I, I) where $D_I \subset Z^2$ is the image domain and $I(p)$ assigns to each pixel $p \in D_I$ a scalar value. The gradient image is interpreted as a graph (D_I, A) whose nodes are the pixels in D_I and whose arcs are defined by an *adjacency relation* A between pixels [39]. We are interested in 4- or 8-connected relations for 2D region-based image segmentation (Figure 2.2a). A *path* in the graph is a sequence of adjacent pixels and a *path-cost function* c assigns to each path π a *path cost* $c(\pi)$.



(a) (b)

Figura 2.2: (a) An 8-connected image graph, where the numbers indicate the pixel values and the object is the shaded square. Internal and external seeds for WS are shown with distinct node patterns (non-black labels). (b) The optimum-path forest, where the numbers indicate minimum costs and distinct labels separate object and background. The arrows link each node to its predecessor in the forest.

Definition 1 (Optimum path) *A path π is* optimum *if $c(\pi) \leq c(\tau)$ for any other path τ with the same destination of π.*

For both WS and TP, the cost $c(\pi)$ of a path is defined as the *maximum gradient value* of its pixels, when π starts in a set S of seed pixels; and as *infinity cost* otherwise.

$$c(\pi) = \begin{cases} \max_{\forall p \in \pi}\{I(p)\} & \text{if } org(\pi) \in S \\ +\infty & \text{otherwise} \end{cases} \tag{2.1}$$

where $org(\pi)$ is the origin of path π. Marker imposition [96, 15, 65] is important in most situations and it is implemented by setting $I(p)$ to 0 for pixels $p \in S$.

The IFT assigns one optimum path (Definition 1) from S to every pixel $p \in D_I$. These paths form an *optimum-path forest* rooted in S which is stored in a *predecessor map* P, such that WS can separate object and background by propagating distinct root labels to their respective trees in the forest (Figure 2.2b).

Definition 2 (Optimum-path forest) *A predecessor map P is a function that assigns to each pixel $p \notin S$ its predecessor $P(p)$ in the optimum path from S or a marker nil when $p \in S$ (in which case p is said to be a root of the forest). An optimum-path forest is a predecessor map which contains no cycles — in other words, one which takes every pixel to nil in a finite number of iterations.*

The IFT algorithm, as presented next, computes at the same time an optimum-path forest in P and a label map in L, being the former useful for TP and the latter applicable for WS.

2.2.2 The IFT algorithm

Let $S = S_o \cup S_b$ be the union of two sets of seed pixels, such that S_o and S_b contain only object and background seeds, respectively. Then, S_b is empty for TP and WS requires S_b not empty.

Algorithm 1 – IMAGE FORESTING TRANSFORM FOR WS AND TP

INPUT:	Gradient image $\hat{I} = (D_I, I)$, adjacency relation A, seed sets S_o and S_b.
OUTPUT:	Optimum-path forest P and label map L.
AUXILIARY:	Cost map C, priority queue Q, and variable cst.

For all $p \in D_I$, set $P(p) \leftarrow nil$ and $C(p) \leftarrow +\infty$.
For all $p \in S_o$, set $C(p) \leftarrow 0$, $L(p) \leftarrow 1$, and insert p in Q.
For all $p \in S_b$, set $C(p) \leftarrow 0$, $L(p) \leftarrow 0$, and insert p in Q.
While Q is not empty, do
    Remove from Q a pixel p such that $C(p)$ is minimum.
    For each q such that $(p,q) \in A$ and $C(q) > C(p)$, do
        Compute $cst \leftarrow \max\{C(p), I(q)\}$.
        If $cst < C(q)$, then
            If $C(q) \neq +\infty$, remove q from Q.
            Set $P(q) \leftarrow p$, $C(q) \leftarrow cst$, $L(q) \leftarrow L(p)$.
            Insert q in Q.

Algorithm 1 runs in linear time when Q is implemented as described in [43]. Lines 1–3 initialize maps and insert seeds in Q. The main loop computes an optimum path from S to every pixel p in a non-decreasing order of cost (Lines 4–11). At each iteration, a path of minimum cost $C(p)$ is obtained in P when we remove its last pixel p from Q (Line 5). Ties are broken in Q using first-in-first-out (FIFO) policy. That is, when two optimum paths reach an ambiguous pixel p with the same minimum cost, p is assigned to the first path that reached it. The rest of the lines evaluate if the path that reaches an adjacent pixel q through p is cheaper than the current path with terminus q and update Q, $C(q)$, $L(q)$ and $P(q)$ accordingly.

The label propagation in L assigns 1 to pixels that belong to the trees rooted inside the object and 0 to pixels of the trees rooted in the background. In WS, it is expected that the object be defined by image components with label 1, which can be directly obtained from L (Figures 2.1d and 2.2b).

Clearly, WS solves segmentation by seed competition for object and background pixels. It also allows simultaneous multiple object segmentation by modifying Algorithm 1 to propagate a distinct label per object. TP requires the identification of leaking pixels in an optimum-path forest P with no external seeds. The object is obtained by pruning all subtrees rooted in the background (Figure 2.1b).

The term "subtree of a node" is used in several parts of the text. A subtree of a node p is a tree rooted at a child node q (i.e., $P(q) = p$), which is obtained by removing from P the arc (p, q).

2.3 Tree-pruning segmentation

Figure 2.3a shows the same image of Figure 2.2a, except that the image's border is not used as external marker. Therefore, the optimum-path forest connects object and background through the leaking pixel $(5, 6)$ [1] in Figure 2.3b. The object can be obtained by removing the subtrees of this leaking pixel.

Note that the IFT algorithm computes optimum paths in a non-decreasing order of costs. Therefore, the optimum paths from S_o will reach object pixels before background pixels whenever the gradient condition for TP (Section 2.2) is satisfied. Moreover, if a pixel p is the only one with lowest value $I(p)$ on the object's boundary, then all pixels around the object will be reached by leaking paths with minimum cost $I(p)$, which pass through the leaking pixel p. By connectivity, the rest of the background will be also conquered by leaking paths that pass through p. When the gradient condition is not fully satisfied, the method may still work (Figure 2.4). The same property can be verified

[1] Pixel locations are given as (x, y) pairs and the top left pixel is $(1, 1)$.

when there are multiple leaking pixels, which can be automatically detected for object definition as follows.

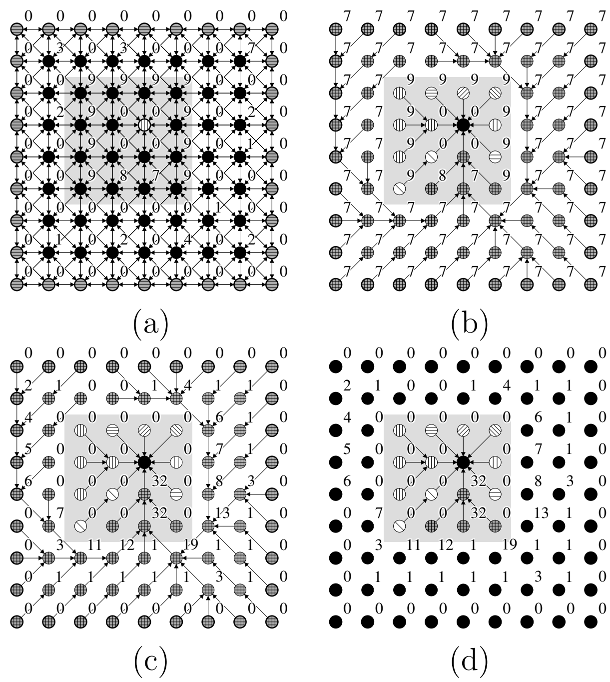


Figura 2.3: (a) The same image graph of Figure 2.2a. Internal seed and pixels in the image's border B are shown with distinct node patterns. (b) Optimum-path forest in TP, where the numbers indicate minimum costs. Object and leaking trees are shown with distinct node patterns. (c) The numbers indicate the descendant count in B for each pixel, except for the root node. (d) After pruning, the remaining forest defines the object.

2.3.1 Object definition

Let $\mathcal{R}$ be the set of the roots in the optimum-path forest (Definition 2). By removing the roots of the forest, we get a forest of subtrees. Each tree of this new forest is classified as being either an *object tree* or as a *leaking tree*.

Definition 3 (Object tree) *An object tree is a subtree of the root nodes, which is contained within the object.*

Definition 4 (Leaking tree) *A leaking tree is a subtree of the root nodes, which also contains optimum paths that reach the background (leaking paths).*

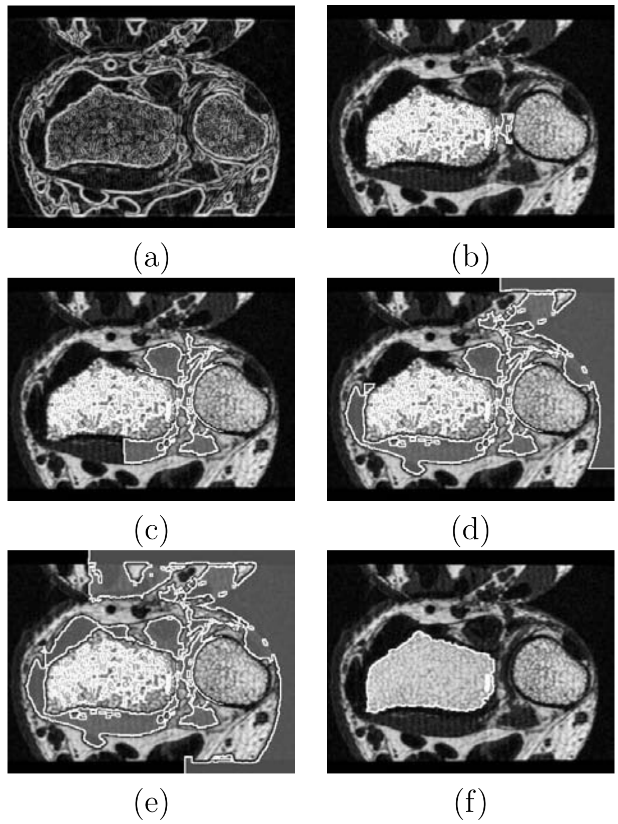


Figura 2.4: (a) A gradient image where the object is a bone of the wrist. (b-e) The region growing of the IFT from internal seeds. The leaking occurs before filling the entire object, but these leaking paths surround the object, avoiding further connections between object and background. (f) The object is obtained by automatic leaking pixel detection.

Figure 2.3b shows several object trees (e.g., one rooted at pixel $(4, 4)$) and a single leaking tree rooted at pixel $(5, 5)$, each one with a distinct node pattern. Let $B \subset D_I$ be the image's border. We compute the number of descendants that every node of the new forest has in B to obtain a *descendant map* D (Figure 2.3c). This combinatorial property of the forest allows the identification of the leaking pixels. The map D is different from the one presented in [37], which computes all descendants in the forest.

Definition 5 (Descendant map) *A descendant map is a function D that assigns to each pixel $p \in D_I \backslash \mathcal{R}$ the number of descendants of p in B.*

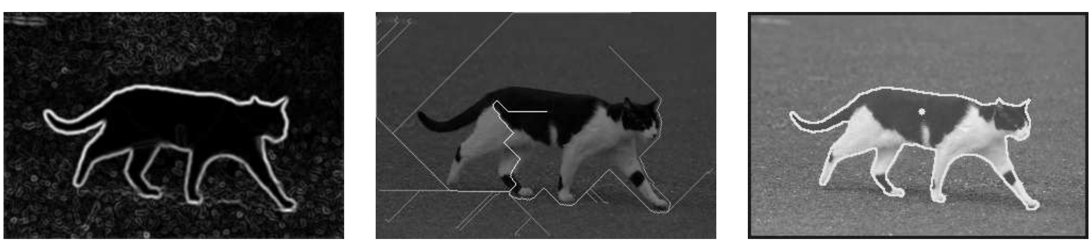
Each leaking tree is supposed to cross the object's boundary through a single pixel, named leaking pixel, and the paths that reach B must pass through all leaking pixels (Figure 2.3b).

Definition 6 (Leaking pixel) *A leaking pixel is defined as an object pixel whose successors in the leaking paths belong to the background.*

The leaking pixels can be usually detected from the predecessor and descendant maps as the *last one* with the highest descendant value along an optimum path that reaches B. That is, by following backwards any optimum path in P, which has terminal node in B, the leaking pixel is the *first one* with the highest descendant value in the way back to the root of its tree (pixel $(5,6)$ in Figure 2.3c). We can repeat this procedure for all nodes in B to detect all leaking pixels automatically.

This property usually holds at the object's boundary thanks to the FIFO tie-breaking policy of the priority queue Q. After leaking, the gradient condition makes ambiguous the pixels in the neighborhood outside the object, ramifying the leaking path into several branches (Figure 2.5). This ramification drastically reduces the descendant count for pixels of the subtrees rooted at the leaking pixels. The object is finally obtained by removing the subtrees of the leaking pixels from the original forest (Figures 2.3d, 2.4f and 2.5c).

Definition 7 (Object by tree pruning) *Let $\mathcal{P}$ be the set of pixels that belong to the subtrees of leaking pixels, $\mathcal{T}_l$ be the set of pixels that belong to the leaking trees, and $\mathcal{T}_o$ be the set of pixels that belong to the object trees. The object is defined as $\mathcal{R} \cup \mathcal{T}_o \cup \{\mathcal{T}_l \backslash \mathcal{P}\}$.*



(a) (b) (c)

Figura 2.5: (a) A gradient image. (b) The original image overlaid by the descendant map. The leaking path ramifies into several branches on the object's boundary, provoking a decreasing in D (the lines become darker) and avoiding further connections between object and background. (c) The resulting segmentation with TP.

2.3.2 Algorithms

Algorithm 2 computes the descendant map D (Definition 5) in linear time. It visits all pixels of the forest in reverse breadth-first order, accumulating the number of descendants from the leaf pixels to the root pixels. Lines 1–3 insert the forest roots in a FIFO queue Q and initialize the descendant count map D. Lines 4–8 traverse the optimum-path forest in breadth-first order, inserting every visited pixel in the stack S. Lines 9–13 use the stack S to visit the forest in reverse breadth-first order, calculating and propagating the descendant count up to the roots of the forest. The resulting map D counts, for each pixel, the number of descendant pixels in the optimum-path forest that belong to B. Algorithm 2 visits each pixel exactly three times, and its running time is $\Theta(|D_I|)$.

Algorithm 2 – DESCENDANT MAP COMPUTATION (LINEAR TIME)

INPUT:	Optimum-path forest P, adjacency relation A, set B.
OUTPUT:	Descendant map D.
AUXILIARY:	Queue Q, Stack S.

1. *For all* $p \in D_I$, *do*
2.   *Set* $D(p) \leftarrow 0$.
3.   *If* $P(p) = nil$, *insert* p *in* Q.
4. *While* Q *is not empty, do*
5.   *Remove a pixel* p *from* Q.
6.   *Insert* p *in* S.
7.   *For each pixel* q *such that* $(p, q) \in A$, *do*
8.   *If* $P(q) = p$, *insert* q *in* Q.
9. *While* S *is not empty, do*
10.   *Remove a pixel* p *from* S.
11.   *If* $P(p) \neq nil$, *then*
12.   *Set* $D(P(p)) \leftarrow D(P(p)) + D(p)$.
13.   *If* $p \in B$, *set* $D(P(p)) \leftarrow D(P(p)) + 1$.

Algorithm 3 computes the same descendant map D with a different approach: For each pixel in B, the algorithm follows backwards its optimum-path up to the root, updating the descendants-in-the-border count. In the worst case – an unlikely situation in which all pixels of B belong to a same optimum-path that contains all image pixels – Algorithm 3 will visit each pixel $|B|$ times, leading to $O(|D_I|\,|B|)$ performance. For 2D images, $|B| \propto \sqrt{|D_I|}$, and its running time becomes $O(|D_I|^{\frac{3}{2}})$. For 3D images, $|B| \propto \sqrt[3]{|D_I|}^2$, and its running time becomes $O(|D_I|^{\frac{5}{3}})$.

Algorithm 3 – DESCENDANT MAP COMPUTATION (ALTERNATIVE)

INPUT:	Optimum-path forest P and set B.
OUTPUT:	Descendant map D.

1. *For all* $p \in D_I$, *set* $D(p) \leftarrow 0$.
2. *For each pixel* $p \in B$, *do*
3. *Set* $q \leftarrow p$.
4. *While* $P(q) \neq nil$, *do*
5. *Set* $D(P(q)) \leftarrow D(P(q)) + 1$, *and* $q \leftarrow P(q)$.

In real applications, Algorithm 3 visits much less than $|D_I|$ pixels since most pixels belong to optimum paths that do not reach set B, and thus they are never visited. While Algorithm 2 guarantees linear performance, Algorithm 3 leads to reduced running times in practical applications. Experimental comparison of these algorithms in real 3D segmentation applications showed that Algorithm 3 is 1.8 times faster than Algorithm 2. The implementation of Algorithm 3 is also shorter and simpler than the implementation of Algorithm 2.

The descendant map D is used to detect the leaking pixel associated with each node in B. For every backwards path from B to its root, the first pixel with the highest value in D along the backwards path is a leaking pixel. The set L_k of leaking pixels is computed by Algorithm 4.

Algorithm 4 – LEAKING PIXEL DETECTION

INPUT: Optimum-path forest P, descendant map D, and set B.
OUTPUT: Set L_k of leaking pixels.

1. *For each pixel* $p \in B$, *do*
2. *Set* $q \leftarrow p$, *set* $d_{max} \leftarrow -\infty$.
3. *While* $P(q) \neq nil$, *do*
4. *If* $D(q) > d_{max}$ *Then set* $d_{max} \leftarrow D(q)$, $r \leftarrow q$.
5. *Set* $q \leftarrow P(q)$.
6. *Set* $L_k \leftarrow L_k \cup \{r\}$.

2.4 Sufficient conditions and geometrical issues

This section discusses the main aspects to understand the differences between watershed transform (WS) and tree pruning (TP), their advantages and limitations. In all examples, we use the same gradient image and internal seeds S_o for WS and TP. Additionally, WS uses the image's border as external seeds S_b and TP uses the image's border B to extract the descendant map D. Therefore, the role of the image's border is very different in these approaches.

The gradient conditions are desirable but not necessary conditions (Section 2.2). In WS, it is not difficult to see that the optimum paths from S_o and S_b will meet at the object's boundary, regardless of seed location, whenever the gradient values are strictly higher on the object's boundary than inside and outside the object. Indeed, WS will work even with one external seed and one internal seed. The gradient condition for TP is more relaxed, but its sufficient conditions are:

(C1) All subtrees of leaking pixels must belong to the background.

(C2) Each leaking tree may have at most one leaking pixel.

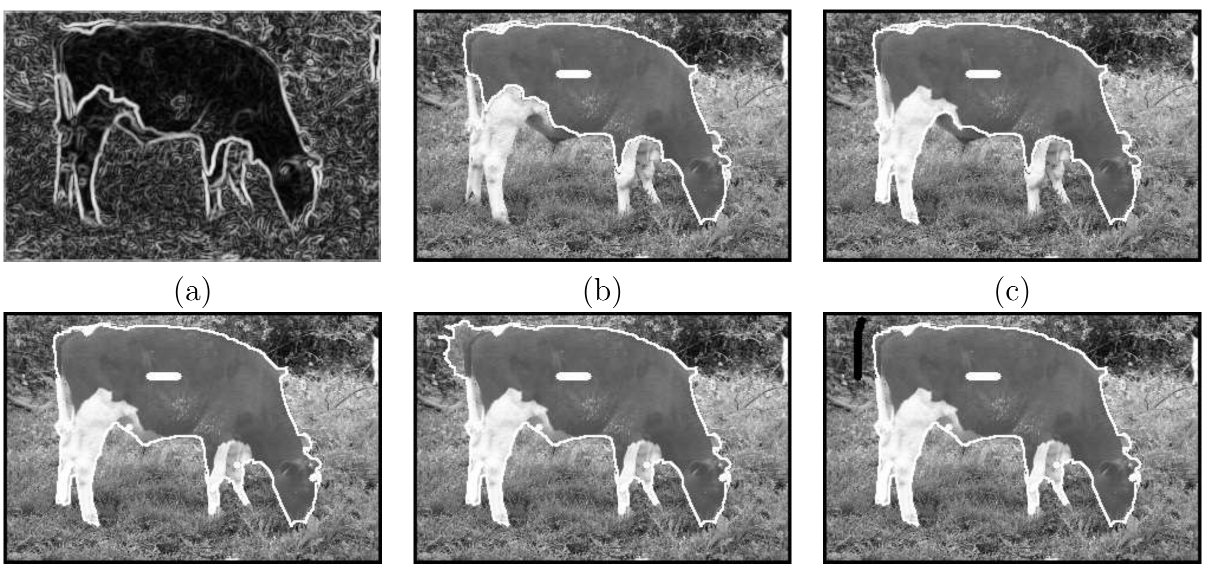
(C3) Each leaking pixel must have at least two children nodes with descendants in B.

The object by Definition 7 is a set of pixels resulting from the union of the roots, object trees, and leaking trees after removing the subtrees of all leaking pixels (background). By definition, roots and object trees belong to the object. Therefore, conditions (C1)–(C3) are sufficient to guarantee that object pixels will not belong to subtrees of leaking pixels and Algorithm 4 will detect all leaking pixels. If (C1) is violated, then object pixels may belong to the subtree of some leaking pixel (because some leaking path left and then returned to the object or a leaking pixel also belongs to some non-leaking path). Condition (C2) guarantees that Algorithm 4 will not detect the highest descendant count inside the object (a false leaking pixel). If a leaking tree has two or more leaking pixels, then the respective leaking paths must have common ancestor nodes inside the object with descendant count strictly higher than the leaking pixels. However, (C2) does not prevent Algorithm 4 to detect a false leaking pixel outside the object. Then (C3) together with (C2) guarantee that all leaking pixels are detected by Algorithm 4. (C3) implies by induction that all leaking pixels have descendant count strictly higher than any of their descendant nodes. Given that the only entrance to the object is through a leaking pixel, the first highest descendant count is detected on the object's boundary for any backwards path from B. Therefore, conditions (C1)–(C3) guarantee the correctness of the method. On the other hand, if the method segments the object then it is not difficult to prove that conditions (C1)–(C3) hold.

Condition (C1) may hold even when the gradient condition for TP is not satisfied (Figure 2.4), but the gradient condition implies in (C1). This can be easily verified because the IFT algorithm computes optimum paths in a non-decreasing order of costs with FIFO tie-breaking policy (Section 2.2.2). This guarantees that under the gradient condition all object pixels will be reached by optimum paths before the background pixels and pixels on the object's boundary will be reached by optimum paths from interior pixels before other boundary pixels.

In interactive segmentation, we may correct the results of TP by adding seeds to S_o whenever (C1) and (C2) fail. This forces object pixels to be reached before background pixels in (C1) and breaks leaking trees with multiple leaking pixels into object trees and/or trees with a single leaking pixel each. Similarly (C3) can be satisfied when a leaking pixel has no descendants in B, by adding pixels of its subtrees to set B. In the worst case, the predecessor of the leaking pixels are added to S_o and their children nodes are added to B, forcing the first highest descendant value to be at the leaking pixel. In [37], the descendant map highlights all leaking paths and not only those that reach B. This is better for 2D interactive segmentation, because the user can see exactly where the leaking occurs and cut the leaking paths.

Figures 2.6a–d illustrate the case that requires more internal seeds to satisfy (C1) and (C2) in TP. The same example does not satisfy the gradient condition for WS. Figures 2.6e–f show that, besides the image's border, additional external seeds are needed in S_b to correct segmentation in WS. Figures 2.7a–b illustrate the case that requires additional external pixels in B to satisfy (C3). The same example shows in Figures 2.7c–d that WS may require more external seeds in S_b for correction than the pixels in B.

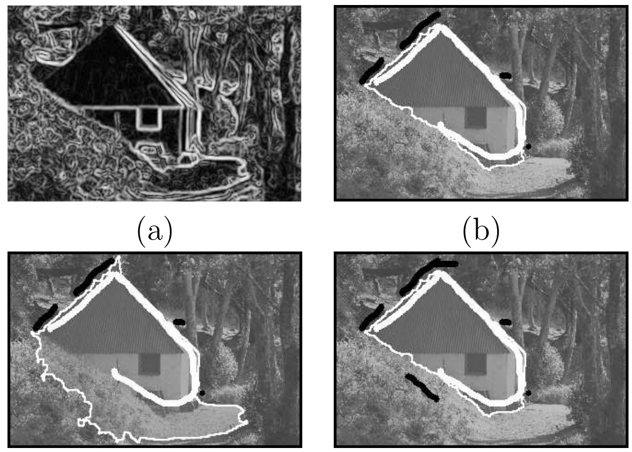


(a) (b) (c)

(d) (e) (f)

Figura 2.6: (a) A gradient image. (b–d) Results of segmentation with TP and incremental seed sets (white markers). (e–f) WS requires the image's border and additional external seeds (black markers) to work.

On the other hand, Figure 2.8 suggests that TP may be more sensitive to the location of the internal seeds than WS. TP fails in Figure 2.8c due to violation of (C2) (the dot in the bigger fragment is a disk with multiple pixels in B). One seed close to the leaking pixel corrects segmentation.



(a) (b) (c) (d)

Figura 2.7: (a) A gradient image. (b) Resulting segmentation with TP, internal seeds (white markers) and additional pixels in B (black markers). (c-d) Resulting segmentation with WS and additional external seeds, besides the image's border (black markers).

Note that in the gradient condition for WS, both methods should provide similar results. We will show this experimentally in Section 2.6.2. Condition (C1) is satisfied as described above and the lower gradient values outside the object together with the FIFO tie-breaking policy favor (C3), as explained in Section 2.3.1. In order to favor (C2), either we select seeds close to the leaking pixels or, in the case of automatic segmentation, estimate as large as possible the seed set.

2.4.1 Heterogeneity of the background

The examples in Figures 2.1, 2.6, and 2.7 suggest that TP may be more robust than WS with respect to the heterogeneity of the background. Figure 2.9a shows a gradient image of a circular object, which satisfies the gradient condition for TP but not for WS. Ambiguous regions are shown in Figure 2.9b as white pixels. These regions are plateaus of cost (tie zones in WS) whose pixels are reached by optimum paths with costs greater than or equal to the leaking pixel values. The watershed lines might be anywhere on these plateaus (Figure 2.9c). Given that TP does not depend on the costs of the optimum paths outside the object, it is less susceptible to the heterogeneity of the background (Figure 2.9d).

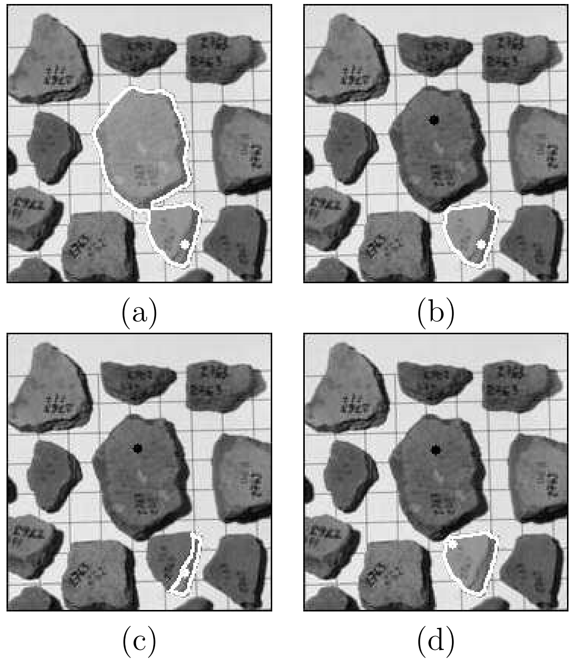


Figura 2.8: (a) Result of segmentation with TP. WS obtains similar result with the image's border as external marker. (b) WS segments the object with one additional external seed (black dot). (c) TP fails with the same seed selection. (d) TP works when we change the location of the internal seed (white dot).

The variant of maximum gradient: A problem may occur when multiple objects with similar gradient values are very close to each other. The absence of space in between the objects to ramify the path at the leaking pixel may create a false pruning point outside the object (Figure 2.10a), failing condition (C3). If the gradient condition for TP is satisfied, we may assume that the correct location of a leaking pixel is always at the maximum gradient value in the path segment between the detected point and its root. We call this *variant of maximum gradient*. Note that it does not affect the location of the true leaking pixels, but solves the problem as illustrated in Figure 2.10b. This variant has been used in all examples and experiments of this paper.

2.4.2 Geometrical issues

The failure probability of the TP conditions increases with the number of leaking pixels. The competition among seeds might prevent some leaking trees to reach set B, failing (C3), and leaking trees might have multiple leaking pixels, failing (C2). Although the gradient condition for TP is not necessary, these observations and its implication in (C1)

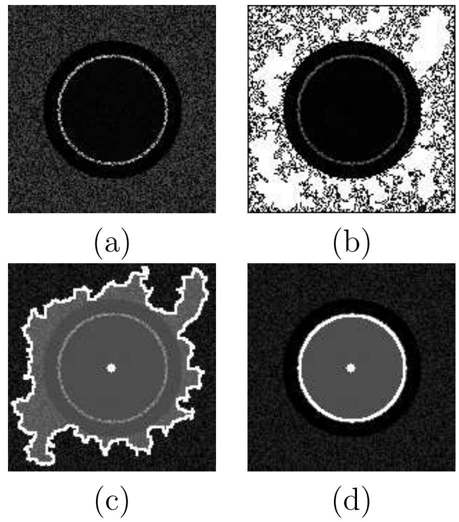


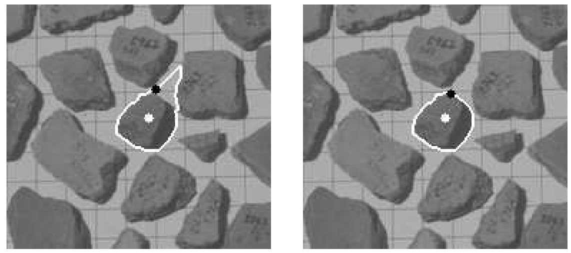
Figura 2.9: (a) A synthetic gradient image of a circular object, where the heterogeneity of the background is given by a random external noise. (b) Watershed tie zones in white. (c-d) Respective segmentation results by WS (c) and TP (d) for a same internal seed (white dot).

indicate that it is at least desirable.

The number of leaking pixels achieves the worst case for perfect boundaries and perfect gaps (Figure 2.11), but there are alternative solutions. Figures 2.11a and 2.11b show a perfect boundary and a wrong segmentation result with TP. A simple solution is to order the pixels by their gradient values such that pixels with the same intensities are randomly ordered (Figures 2.11c and 2.11d). In perfect gaps (Figure 2.11e), we can add internal seeds in S_o and external pixels in B around the gaps to solve segmentation (Figure 2.11f). A similar solution is required for WS. However, a more elegant solution is to run an edge detection method, compute the Euclidean distance transform to the edges, and create an edge distance map as the complement of the pixel distances to the edges. By adding the edge distance values to the original gradient value, we obtain a new gradient image in which the number of leaking pixels is drastically reduced (Figure 2.11g), solving the problem with perfect gaps (Figure 2.11h).

Fortunately, images usually have noise that considerably reduces the number of leaking pixels. Therefore, we never needed to use these solutions in practice.

Condition (C3) may also fail in the case of nested boundaries, being them a problem for WS as well. External (true or false) closed boundaries may prevent leaking paths of internal boundaries to reach set B. The problem can be solved with an alternative pixel in B, which should be selected between the boundaries. Similar solution can be obtained



(a) (b)

Figura 2.10: Image with one object fragment marked for detection. (a) The detected pixel is outside the fragment due to its proximity to the other fragments. (b) The correct leaking pixel is automatically detected using the variant of maximum gradient.

with S_b in WS. In [69], we proposed an alternative solution for TP which iteratively searches the desired boundary (leaking pixel) by matching candidates with a template. This solution goes backwards along the optimum path from the first detected pixel to its root, looking for a next pixel whose gradient value is maximum in the remaining segment. A third and probably best option is to improve gradient computation in order to avoid such situations (Figure 2.12).

2.5 Gradient images

It should be clear that under the gradient condition for WS, TP and WS provide similar results, except for some pixels on the object's boundary because WS divides the boundary between influence zones of internal and external seeds. In order to create suitable gradient images for both methods, we should be able to exploit image features which distinguish object and background.

Let $\vec{f}(p) = (f_1(p), f_2(p), \ldots, f_n(p))$ be a vector at pixel p such that the values $f_i(p)$, $i = 1, 2, \ldots, n$, are the brightness values of p and of its 8-neighbors, for instance. The differences $d_j(p, q_j)$, $j = 1, 2, \ldots, 8$, measure the brightness variations around arcs (p, q_j) between p and each of its 8-neighbors.

$$d_j(p, q_j) = \frac{1}{n} \sum_{i=1}^{n} f_i(q_j) - f_i(p). \qquad (2.2)$$

The vector $\vec{d_j}(p, q_j) = d_j(p, q_j) \frac{q_j - p}{|q_j - p|}$ represents the brightness variation in the direction of (p, q_j). The gradient vector $\vec{G}(p)$ at p is obtained by the sum of its projections on these

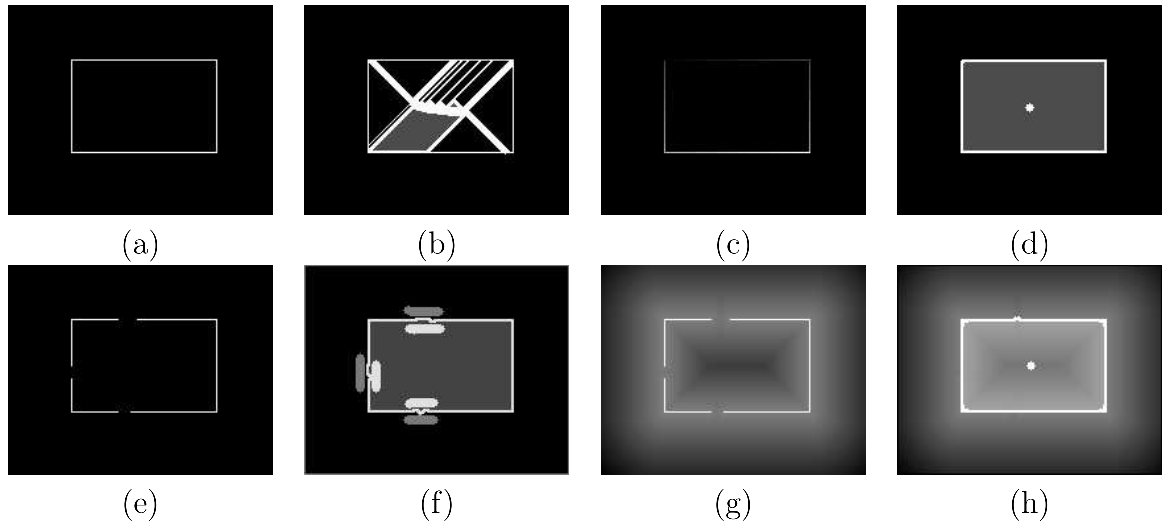


Figura 2.11: (a) Synthetic gradient with perfect boundary. (b) TP fails because it missed several leaking pixels. (c) A randomly ordered gradient. (d) TP works on (c) with a single seed. (e) Synthetic gradient with perfect gaps. (f) A hard solution using markers around the gaps. (g) A gradient image with edge distance map. (h) TP works on (g) with a single seed.

directions.

$$\vec{G}(p) = \sum_{j=1}^{8} \vec{d}_j(p, q_j). \tag{2.3}$$

The magnitude of $\vec{G}$ is used as gradient image $\hat{I}$. In the case of colored images (the originals of Figures 2.6, 2.7, 2.8, and 2.10), we obtain one gradient image for each channel; red $\hat{I}_r$, green $\hat{I}_g$, and blue $\hat{I}_b$; and compute the final gradient value $I(p) = \max\{I_r(p), I_g(p), I_b(p)\}$ for all pixels $p \in D_I$.

2.6 Evaluation

TP and WS require some approach for seed selection (object location) in automatic segmentation. We present two applications in which robust procedures exist for object location: (i) license plate segmentation and (ii) 3D MR-image segmentation of the human brain.

The experiments compare the results with some ground truth, which was obtained by interactive differential watershed segmentation [40] for both applications. Let O and G be

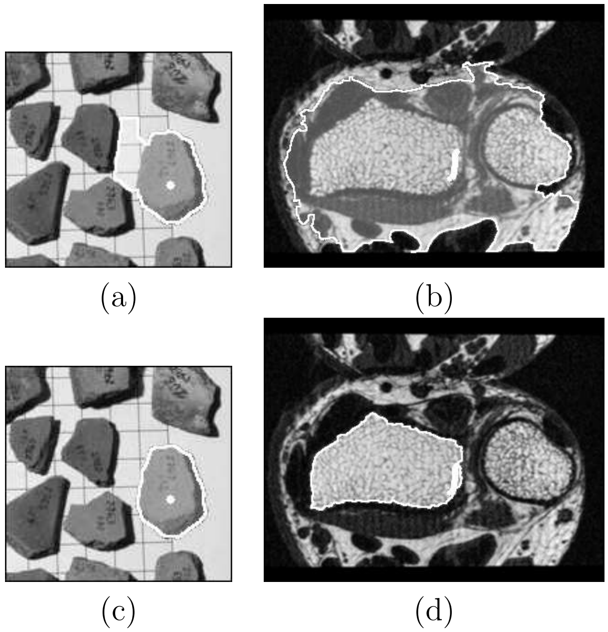


Figura 2.12: (a-b) Results of TP for a fragment and a bone of the wrist using the magnitude of the Sobel's gradient and (c-d) the improved gradient presented in Section 2.5.

the pixel sets that represent segmented object and ground truth. We use three normalized measurements: *false negatives* (FN), *false positives* (FP), and *error* (E).

$$FN = \frac{|G \setminus O|}{|G|} \quad (2.4)$$

$$FP = \frac{|O \setminus G|}{|O|} \quad (2.5)$$

$$E = \frac{|(O \setminus G) \cup (G \setminus O)|}{|O \cup G|} \quad (2.6)$$

where $|X|$ is the cardinality of the set X.

2.6.1 License Plate Segmentation

The experiments used 990 images (352×240 pixels) from a database of license plates. The goal is to find the precise location and spatial extent of the plates (Figure 2.13a). Seed selection is a difficult task, because any attempt to estimate seeds inside a plate is likely to find seeds in other parts of the image. Besides, we need to estimate seed pixels outside the plate numbers to avoid problems with nested boundaries (Figure 2.13b).

(a) (b)

Figura 2.13: (a) Original image with the result of TP. (b) The magnitude of the Sobel's gradient.

Seed selection and gradient image

For automatic seed selection, we chose a method proposed by Zheng et al [101]. This approach is very effective for plate location, but could not be used as baseline for segmentation. The reason is that they ignore shape deformations which do not occur in their database, but are very common in ours (Figure 2.14a). The magnitude of the Sobel's operator was chosen as gradient image. This choice illustrates our observation that WS is more dependent of the heterogeneity of the background than TP.

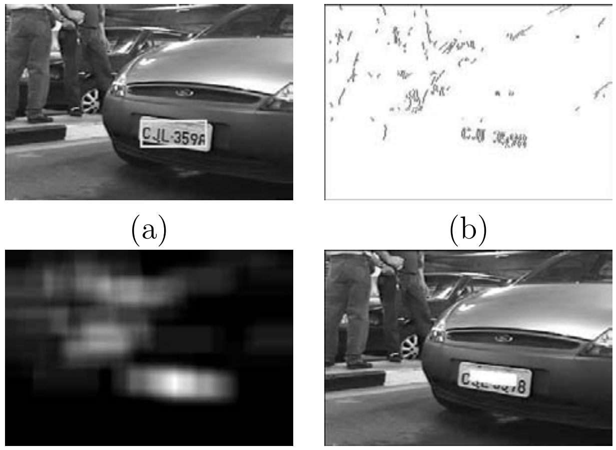
The original image is first enhanced, then vertical edges are extracted using Sobel's operator (Figure 2.14b). An edge density map is computed using a rectangular window (Figure 2.14c). The center of the plate is expected to be at the highest value in this map.

Seed estimation around the point detected by Zheng's approach is done by "supervised learning". The training set consists of 15% of the plates randomly selected from the database. We compute the average of their binary images (ground truths), with all plates translated to a same reference point (a common geometric center). The pixels with average value 1 are interior pixels in these plates, and so they are assumed to be interior pixels of the remaining plates in the database. To avoid seeds on the border of the plates, their erosion with a disk of radius 1 is used as internal seed set for TP and WS (Figure 2.14d). Pixels of the image's border are used as external seeds S_b for WS and set B for TP.

The seed estimation based on Zheng's approach found seed sets inside 93.3% of the plates in our database. We could further improve this result to 96.4% by taking into account the fact that there is no license plate in the top region of the image (30% of the height) in our database.

Experiments and results

We compare the segmentation results of TP, WS, and a previous version of TP [37] (PTP— previous TP), which requires a parameter T for leaking pixel detection. In order



(a) (b) (c) (d)

Figura 2.14: (a) Original image with the result of the method proposed by Zheng. (b) Vertical edges after filtering. (c) The edge density map. (d) Internal seed set.

to show robustness with respect to the seed estimation process, we executed segmentation 10 times for each method, using a distinct training set each time, and computed mean and standard deviation of the average values of E, FN, and FP (Equations 2.4-2.6). Table 2.1 summarizes the results of each method using as test set the images where seed estimation was inside the plates (i.e., 96.4% of the 990 plates).

Tabela 2.1: License plate segmentation: Mean and standard deviation of the average values of error (E), false negatives (FN) and false positives (FP) for each method with respect to the ground truth.

Method	**Error (E)**	**False Negatives (FN)**	**False Positives (FP)**
TP	2.72% ± 0.07%	0.89% ± 0.09%	1.90% ± 0.10%
WS	4.84% ± 0.10%	2.09% ± 0.07%	2.85% ± 0.11%
PTP(T=1%)	5.85% ± 0.74%	3.34% ± 0.76%	2.61% ± 0.03%
PTP(T=3%)	4.15% ± 0.07%	0.56% ± 0.06%	3.66% ± 0.10%
PTP(T=5%)	5.37% ± 0.05%	0.56% ± 0.02%	4.90% ± 0.06%

Note that TP was more accurate than PTP for any fixed value T, because it is difficult to adjust T in this application. Figures 2.15a-b illustrate cases of errors in seed estimation. In some cases Zheng's method detects a point outside the plate and in other cases the detected point is far from the center of the plate (as consequence, internal seeds are wrongly estimated outside the plate). Table 2.1 also shows that WS was more sensitive than TP with respect to the heterogeneity of the background (Figures 2.15c-d). Figures 2.15e-i show correct segmentation results using TP under various conditions.

If we consider the maximum error $E = 0.10$ (10%) as an acceptable segmentation, 93% of the 990 plates are detected with TP and 91% are detected with WS.

Deformable models [59, 26] could also be used to segment the plates from the detected location. However, the edges inside and outside the plates might present a local minima problem (Figure 2.13b) and the model should take into account shape deformations and imperfect plates (Figures 2.15e-h). In this case, it seems at least simpler to use approaches like TP and WS.

In comparison with the TP-based plate segmentation method presented in [69], the current version gives similar results but works faster (55 msec per image on an Athlon XP 2400+) using a more elegant procedure for seed estimation and a single tree pruning execution per image.

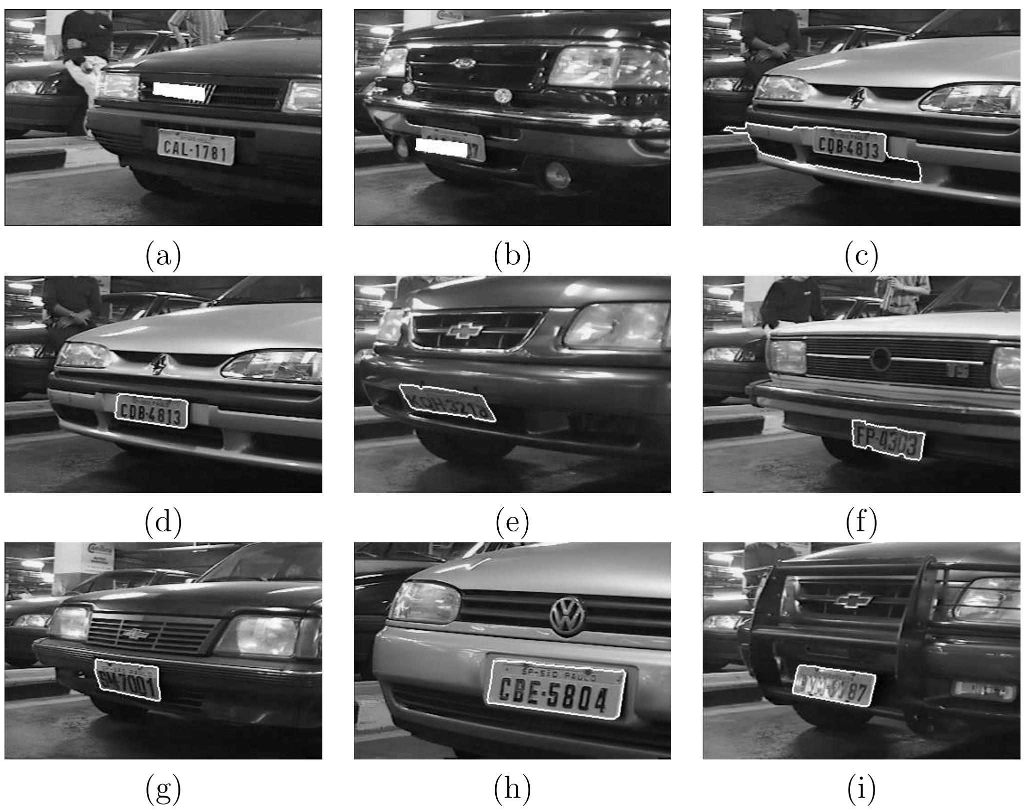


Figura 2.15: (a-b) Errors in the seed selection procedure. (c) Watershed fails segmentation. (d) Tree pruning correctly segments the same image in (c). (e-i) License plate segmentations by TP under shape distortions, scale changes and lighting effects.

2.6.2 3D MR-image segmentation of the brain

Segmentation of the human brain from Magnetic Resonance (MR) images has been addressed in several different ways, each one with its pros and cons [50]. Figure 2.16 shows an MR-T1 slice image of the brain, where some structures and tissues are indicated. The goal is to separate the gray matter (GM) and white matter (WM), as a single object, from the rest of the image. We have evaluated tree pruning using phantom MR-T1 images (which are available at the BrainWeb site[2] [21] together with their ground truth) and real MR-T1 images from control subjects.

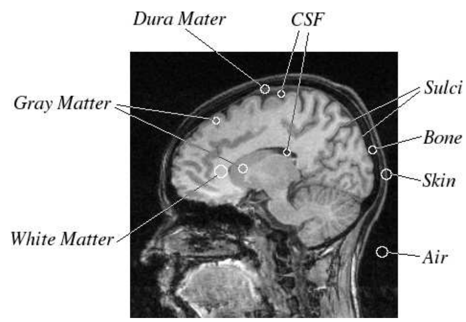


Figura 2.16: Tissues in MR-T1 images of the brain. CSF is the cerebrospinal fluid (liquor).

Seed selection and gradient image

Figure 2.17a shows an MR-T1 slice image of the brain. Note that GM and WM can not be obtained by automatic thresholding on Figure 2.17a and simple morphological operations on Figure 2.17b. On the other hand, a morphological erosion on Figure 2.17b with a sphere of radius 5mm can assuredly separate GM and WM from other structures, and the largest connected component resulting from erosion can be used as a seed set inside the brain (Figure 2.17c). The gradient image is computed as described in Section 2.5, but using a three-dimensional 6-neighborhood instead of the two-dimensional 8-neighborhood (Figure 2.17d). This choice illustrates our observation that WS and TP produce similar results when the gradient condition for WS is satisfied.

Experiments with phantoms

On the first set of experiments, we generated 8 MR-T1 phantoms with varying noise and inhomogeneity (INU) settings (Figure 2.18), and automatically segmented them with tree

[2]URL: http://www.bic.mni.mcgill.ca/brainweb/

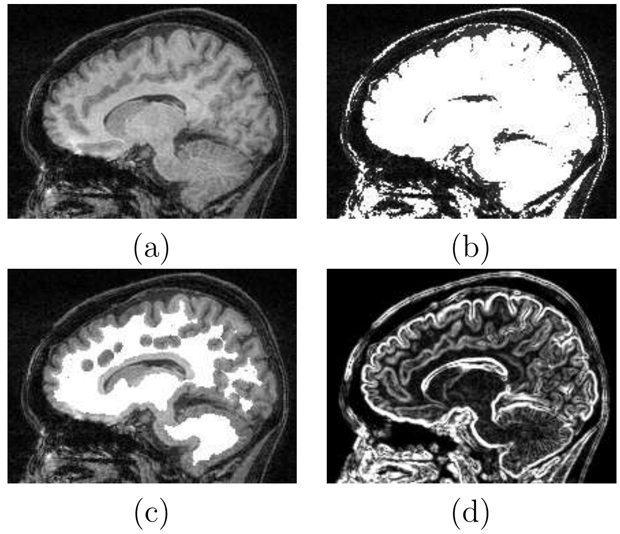


Figura 2.17: (a) Sample slice of MR-T1 image of the brain. (b) Binary slice resulting from Otsu's thresholding overlaid on image (a). (c) Binary slice of seed voxels overlaid on image (a). (d) Slice of the gradient image.

pruning (TP) and watershed (WS), using the same internal seeds, gradient image, and the image's border as background seeds for WS and set B for TP. The results were compared with the available ground truth.

Since TP and WS detect the external boundary of the brain, the resulting objects still contain small CSF regions (sulci and ventricles). To properly classify them as background, we take the intersection between the pruned object and the voxels with intensities above the Otsu's threshold.

Experiments with real MR-T1 images

On the second set of experiments, we selected 20 MR-T1 images of control subjects with no known anomalies, acquired with a 2T Elscint scanner and voxel size of $0.98 \times 0.98 \times 1.00$ mm^3. All volumes were interactively segmented with differential watersheds [40] in order to provide a basis for comparison (a ground truth). The images were automatically segmented by 3 methods: TP, WS (with the image's border as external seeds), and SPM2 [47]— a widely used template-based approach for medical research.

Results

Table 2.2 shows the segmentation errors of TP and WS for the phantoms with respect to the available ground truth. Figure 2.19 shows 3D renditions of the obtained segmentations.

Tabela 2.2: Brain phantom segmentation errors with tree pruning and watershed.

	Tree Pruning	**Watershed**
Noise / INU	Error (FN,FP)	Error (FN,FP)
3% / 20%	5.16% (3.20%,2.09%)	4.58% (2.53%,2.15%)
5% / 20%	5.03% (3.14%,2.01%)	4.41% (2.55%,1.95%)
7% / 20%	5.26% (3.65%,1.74%)	4.69% (2.96%,1.84%)
9% / 20%	6.00% (4.48%,1.66%)	5.42% (3.85%,1.70%)
3% / 40%	5.26% (3.16%,2.23%)	4.61% (2.45%,2.27%)
5% / 40%	5.18% (3.44%,1.86%)	4.70% (2.73%,2.08%)
7% / 40%	5.67% (4.06%,1.75%)	5.09% (3.38%,1.83%)
9% / 40%	6.68% (5.23%,1.61%)	6.14% (4.66%,1.64%)
Average	5.53% (3.80%,1.87%)	4.96% (3.14%,1.93%)

Table 2.3 shows the segmentation errors for the 3 methods on real images, using the interactive segmentation as the basis for comparison. Figure 2.20 shows renditions of the segmented brains.

We can observe the high degree of agreement among TP, WS and the ground truths in both experiments. Note that SPM2 provided segmentations with noticeable artifacts, such as disconnected components outside the brain (Figure 2.20). This illustrates how difficult is to segment our images.

TP and WS not only provided better segmentation results than SPM2 on the real images, but also were much faster. The entire segmentation task with TP (marker extraction, gradient computation, optimum-path forest computation, leaking point detection and tree pruning), for example, took 25 seconds for a $181 \times 217 \times 181$ MR-T1 volume, while SPM2 took 5 minutes, on the same Athlon64 3200+ workstation.

The background is homogeneous in most part of these MR-brain images and there are no strong barriers between the brain's border and the image's border. This favored gradient images that satisfy the gradient condition for WS. As we expected, such a situation makes TP and WS to provide similar results. The segmentation differences were mostly the segmentation of excess tissue (false positives) near the cerebellum (as seen in Figure 2.20 for subjects 9, 17 and 20) and near the optical nerve (as seen in Figure 2.20 for subjects 6, 8, 9 and 19), as well as a thin layer of misclassified voxels over the brain's cortex, indicating off-by-one errors in the location of the object's boundary. These mistakes can be minimized with the design of better application-dependent methods for the computation of the gradient image, with stronger barriers between object and background in the problematic regions. In the case of real images, the slight advantages of WS in

Tabela 2.3: Real brain segmentation errors with tree pruning, watershed and SPM2.

Subject	**Tree Pruning** Error (FN,FP)	**Watershed** Error (FN,FP)	**SPM2** Error (FN,FP)
01	6.63% (2.88%, 2.69%)	5.62% (2.51%, 2.16%)	10.49% (4.87%, 3.92%)
02	8.79% (1.62%, 6.94%)	8.28% (1.42%, 6.75%)	14.56% (5.85%, 6.77%)
03	7.83% (2.33%, 4.82%)	7.30% (1.80%, 5.08%)	13.00% (4.31%, 7.62%)
04	11.60% (3.46%, 6.07%)	7.62% (3.04%, 2.32%)	15.10% (6.39%, 4.46%)
05	9.85% (2.34%, 6.71%)	10.24% (2.06%, 7.74%)	19.51% (10.21%, 4.04%)
06	9.59% (2.39%, 6.02%)	8.32% (2.05%, 5.27%)	16.47% (4.05%, 10.71%)
07	7.98% (1.55%, 6.17%)	8.24% (1.27%, 6.88%)	11.70% (3.89%, 6.71%)
08	8.37% (3.36%, 3.75%)	7.05% (2.87%, 3.12%)	13.20% (5.37%, 6.15%)
09	9.32% (2.70%, 5.74%)	9.35% (2.33%, 6.45%)	12.36% (5.21%, 5.20%)
10	13.96% (1.80%,12.21%)	13.46% (1.56%,12.05%)	15.50% (5.70%, 7.92%)
11	6.82% (1.82%, 4.67%)	6.77% (1.56%, 4.98%)	12.68% (5.09%, 6.40%)
12	12.88% (2.22%,10.51%)	12.93% (1.82%,11.14%)	15.78% (5.20%, 9.01%)
13	8.51% (1.98%, 6.22%)	8.12% (1.56%, 6.47%)	12.41% (4.62%, 6.56%)
14	8.96% (3.89%, 4.15%)	8.49% (3.43%, 4.27%)	12.36% (6.00%, 5.09%)
15	7.00% (2.07%, 3.96%)	6.02% (1.78%, 3.42%)	11.82% (4.99%, 4.22%)
16	9.71% (2.45%, 6.44%)	12.61% (2.20%,10.25%)	14.70% (5.20%, 7.14%)
17	7.66% (2.94%, 3.51%)	7.30% (2.61%, 3.65%)	12.57% (5.01%, 5.81%)
18	7.15% (2.81%, 2.62%)	6.41% (2.61%, 2.08%)	16.43% (7.92%, 3.93%)
19	9.14% (1.41%, 7.87%)	8.26% (1.29%, 7.00%)	13.01% (4.17%, 8.38%)
20	16.03% (1.85%,14.71%)	14.31% (1.70%,12.97%)	18.34% (3.31%, 14.87%)
Average	9.39% (2.39%, 6.29%)	8.83% (2.07%, 6.20%)	14.10% (5.37%, 6.74%)

performance are also related to the fact that the ground truth was created using the interactive watershed on the same 3D gradient image [40].

2.7 Conclusion

We have presented a considerably extended version of our previous work on tree-pruning segmentation [37, 69], which adds a formal definition of the obtained objects, several new examples, algorithms, sufficient conditions, geometrical issues, an improved gradient computation with respect to the Sobel's operator, and experiments for automatic license plate image segmentation and 3D MR-image segmentation of the human brain. The experiments show that TP can provide good results in both applications, is less sensitive than WS with respect to the heterogeneity of the background, and that both approaches provide similar results when the gradient condition for WS is satisfied.

We may see image segmentation as consisting of two tightly coupled tasks: *recognition* and *delineation*. Delineation aims to define the precise spatial extent of an object in the image while its approximate location (e.g., seed estimation) is a recognition task. Recognition also involves other cognitive tasks, such as to verify the segmentation correctness or to identify a desired object among candidate ones. While computers usually outperform human beings in delineation, the other way around has been verified for recognition [44].

TP and WS are essentially delineation methods. At same time, model-based approaches [26, 25] have been proposed for recognition and delineation. While they are effective recognition methods, it is usually very difficult to model all possible variations of a given object. In this sense, we believe that methods such as WS and TP, which provide delineation based on optimum criteria, should be combined with object location by model-based approaches for automatic segmentation. Our future work follows in this direction.

Acknowledgments

The authors thank Roberto Lotufo (FEEC-UNICAMP) for the images of license plates and Laurent Najman (ESIEE) for his comments on this paper. This work was supported by CNPq (Proc. 302427/04-0), CAPES and FAPESP (Proc. 03/09793-1 and Proc. 03/13424-1).

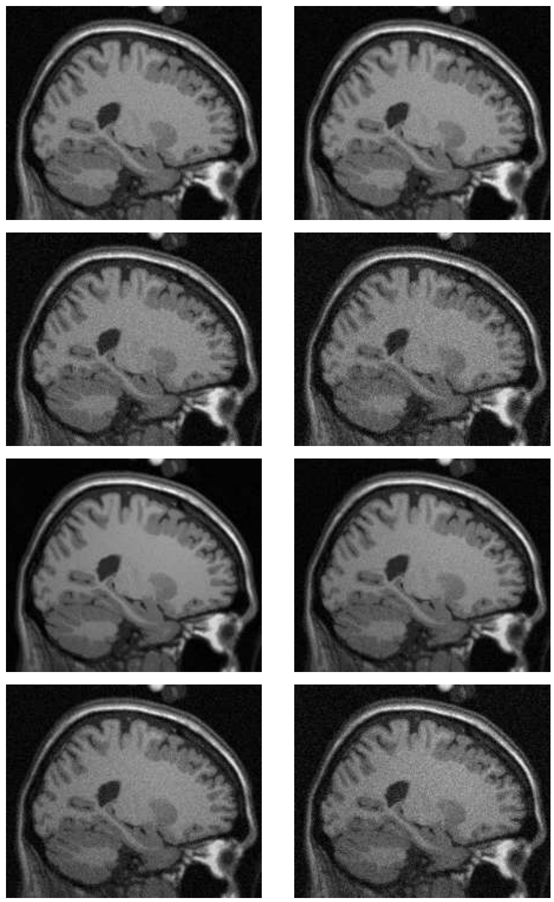


Figura 2.18: Sample slices of the phantoms with different degrees of noise (N) and inhomogeneity (INU): (a) $N = 3\%$ and $INU = 20\%$, (b) $N = 5\%$ and $INU = 20\%$, (c) $N = 7\%$ and $INU = 20\%$, (d) $N = 9\%$ and $INU = 20\%$, (e) $N = 3\%$ and $INU = 40\%$, (f) $N = 5\%$ and $INU = 40\%$, (g) $N = 7\%$ and $INU = 40\%$, and (h) $N = 9\%$ and $INU = 40\%$.

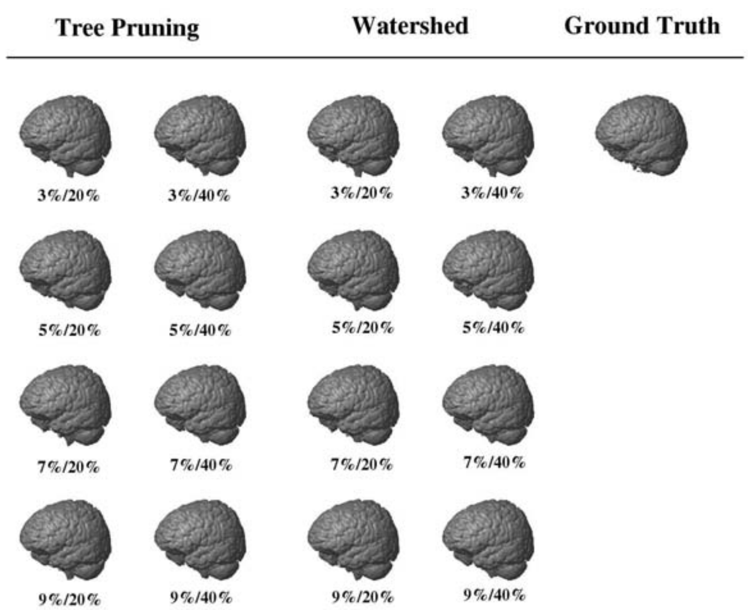


Figura 2.19: Renditions of the 8 brain phantom segmentations by tree pruning (first and second columns), watershed (third and fourth columns) and of the provided ground truth (fifth column).

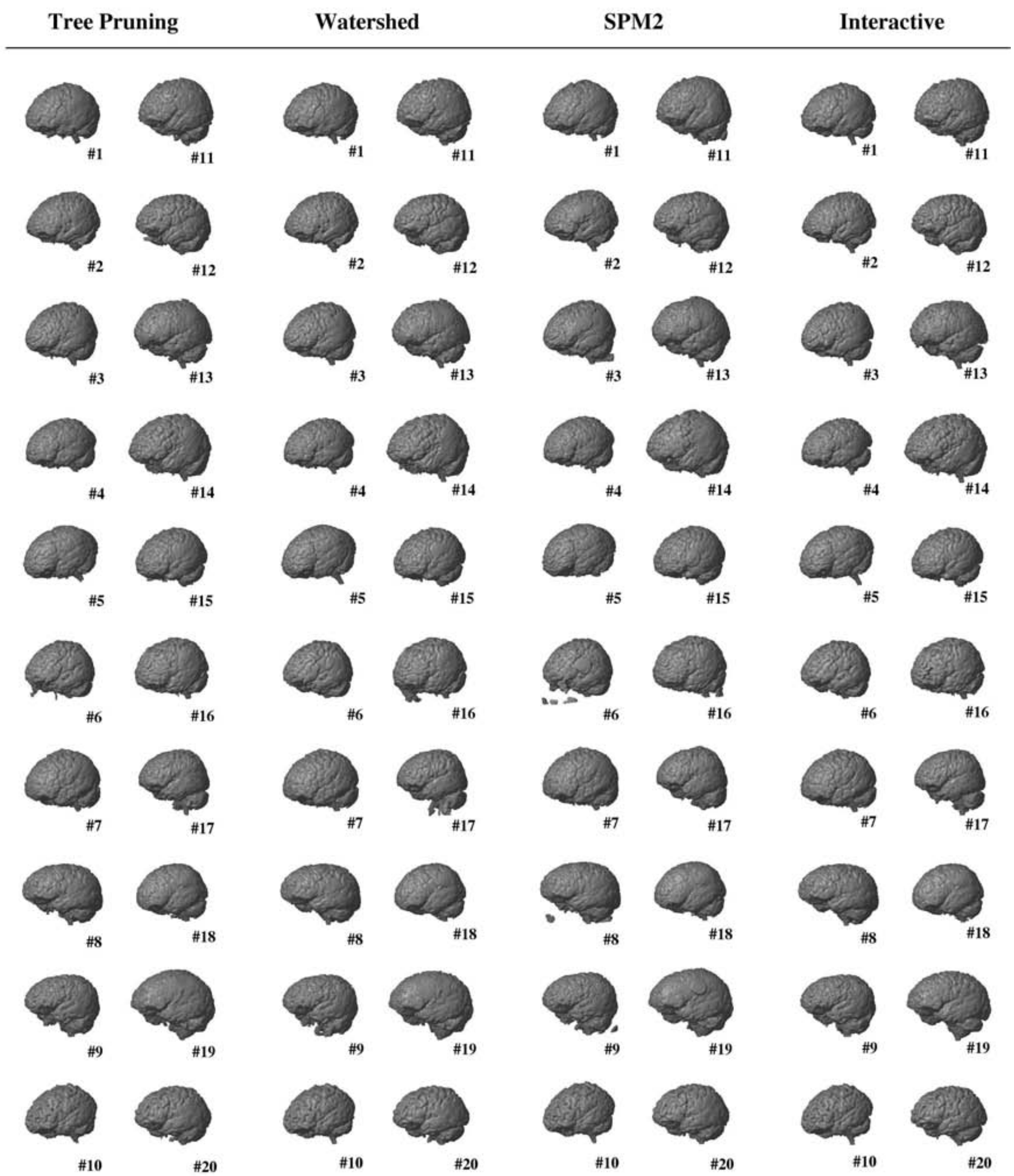


Figura 2.20: Brain segmentation results with tree pruning, watershed and SPM2 for 20 control subjects, shown as 3D renditions.

Capítulo 3

Fast and Robust Mid-sagittal Plane Location in 3D MR Images of the Brain

3.1 Introduction

The human brain is not perfectly symmetric [32, 27, 49]. However, for the purpose of analysis, it is paramount to define and distinguish a *standard of asymmetry*, considered as normal for any given measurement, from abnormal asymmetry, which may be related to neurological diseases, cerebral malformations, surgical procedures or trauma. Several works sustain this claim. For example, accentuated asymmetries between left and right hippocampi have been found in patients with Schizophrenia [99, 28, 87, 67, 54, 6], Epilepsy [55, 100] and Alzheimer Disease [29, 64].

The brain can be divided in two hemispheres, and the structures of one side should have their counterpart in the other side with similar shapes and approximate locations [32]. These hemispheres have their boundaries limited by the longitudinal (median) fissure, being the corpus callosum their only interconnection.

The ideal separation surface between the hemisferes is not perfectly planar, but the mid-sagittal plane (MSP) can be used as a reference for asymmetry analysis, without significant loss in the relative comparison between normal and abnormal subjects. The MSP location is also important for image registration. Some works have used this operation as a first step for intra-subject registration, as it reduces the number of degrees of freedom [3, 58], and to bring different images into a same coordinate system [63], such as in the Talairach [89] model.

However, there is no exact definition of the MSP and its determination by manual delineation is sensitive to different experts. Given that, a reasonable approach for evaluation seems to be visual inspection with error quantification, when we increase the asymmetry artificially and/or linearly transform the image.

The longitudinal fissure forms a gap between the hemispheres filled with cerebro-spinal fluid (CSF). We define the MSP as a large intersection between a plane and an *envelope* of the brain (a binary volume whose surface approximates the convex hull of the brain) that maximizes the amount of CSF. This definition leads to an automatic, robust and fast algorithm for MSP extraction.

The paper is organized as follows. In Section 3.2, we review existing works on automatic location of the mid-sagittal plane. In section 3.3, we present the proposed method. In section 3.4, we show experimental results and validation with simulated and real MR-T1 images. Section 3.5 states our conclusions.

3.2 Related works

MSP extraction methods can be divided in two groups: (i) methods that define the MSP as a plane that maximizes a symmetry measure, extracted from both sides of the image [57, 68, 88, 3, 85, 63, 77, 94, 91], and (ii) methods that detect the longitudinal fissure to estimate the location of the MSP [17, 51, 56, 98]. Table 3.1 summarizes these works, and extensive reviews can be found in [56], [98], [77] and [63].

Methods in the first group address the problem by exploiting the hough symmetry of the brain. Basically, they consist in defining a symmetry measure and searching for the plane that maximizes this score. Methods in the second group find the MSP by detecting the longitudinal fissure. Even though the longitudinal fissure is not visible in some modalities, such as PET and SPECT, it clearly appears in MR images. Particularly, we prefer these methods because patients may have very asymmetric brains and we believe this would affect the symmetry measure and, consequently, the MSP detection.

The aforementioned approaches based on longitudinal fissure detection present some limitations that we are circumventing in the proposed method. In [51], the MSP is found by using snakes and orthogonal regression for a set of points manually placed on each slice along the longitudinal fissure, thus requiring human intervention. Other method [17] uses the Hough Transform to automatically detect straight lines on each slice [17], but it does not perform well on pathological images. The method in [56] assumes local symmetry near the plane, which is not verified in many cases (see Figures 3.2, 3.5 and 3.8). Volkau et al. [98] propose a method based on the Kullback and Leibler's measure for intensity histograms in consecutive candidate planes (image slices). The method presents excellent results under a few limitations related to rotation, search region of the plane, and pathological images.

Tabela 3.1: Summary of existing MSP methods

Method	**Based on**	**2D/3D**	**Application**	**Measure**
[17]	fissure	2D	MR	Edge Hough Transform
[51]	fissure	2D	MR	Active contours
[56]	fissure	2D	MR, CT	Local symmetry of fissure
[98]	fissure	3D	MR, CT	Kullback-Leibler's measure
[57]	symmetry	2D	PET, SPECT	Intensity cross correlation
[68]	symmetry	3D	PET	Stochastic sign change
[3]	symmetry	3D	MR, PET	Intensity cross correlation
[88]	symmetry	3D	MR, CT	Extended Gaussian image
[85]	symmetry	3D	MR, CT, PET, SPECT	Ratio of intensity profiles
[63]	symmetry	2D	MR, CT	Edge cross correlation
[77]	symmetry	3D	MR, CT, PET, SPECT	Intensity cross correlation
[94]	symmetry	3D	MR, CT, SPECT	Intensity cross correlation
[91]	symmetry	3D	MR	Edge cross correlation

3.3 Methods

Our method is based on detection of the longitudinal fissure, which is clearly visible in MR images. Unlike some previous works, our approach is fully 3D, automatic, and applicable to images of patients with severe asymmetries.

We assume that the mid-sagittal plane is a plane that contains a maximal area of cerebro-spinal fluid (CSF), excluding ventricles and lesions. In MR T1 images, CSF appears as low intensity pixels, so the task is reduced to the search of a sagittal plane that minimizes the mean voxel intensity within a mask that disregards voxels from large CSF structures and voxels outside the brain.

The method is divided in two stages. First, we automatically segment the brain and morphologically remove thick CSF structures from it, obtaining a brain mask. The second stage is the location of the plane itself, searching for a plane that minimizes the mean voxel intensity within its intersection with the brain mask. Our method uses some morphological operations whose structuring elements are defined based on the image resolution. To keep the method description independent of image resolution, we use the notation S_r to denote a spherical structuring element of radius r mm.

3.3.1 Segmentation Stage

We use the tree pruning approach to segment the brain. Tree pruning [37, 69] is a segmentation method based on the Image Foresting Transform [39], which is a general tool for the design of fast image processing operators based on connectivity. In tree pruning, we interpret the image as a graph, and compute an optimum path forest from a set of seed voxels inside the object. A gradient-like image with high pixel intensities along object borders must be computed to provide the edge weights of the implicit graph. A combinatorial property of the forest is exploited to prune tree paths at the object's border, limiting the forest to the object being segmented.

To segment the brain (white matter (WM), gray matter (GM) and ventricles), we compute a suitable gradient image, a set of seed voxels inside the brain and apply the tree pruning algorithm. A more detailed description of this procedure is given in [12]. Note that any other brain segmentation method could be used for this purpose.

Gradient computation. MR-T1 images of the brain contain two large clusters: the first with air, bone and CSF (lower intensities), and the second, with higher intensities, consists of GM, WM, skin, fat and muscles. Otsu's optimal threshold [74] can separate these clusters (Figs. 3.1a and 3.1b), such that the GM/CSF border becomes part of the border between them. To enhance the GM/CSF border, we multiply each voxel intensity $I(p)$ by a weight $w(p)$ as follows:

$$w(p) = \begin{cases} 0 & I(p) \leq m_1 \\ 2\left(\frac{I(p)-m_1}{m_2-m_1}\right)^2 & m_1 < I(p) \leq \tau \\ 2 - 2\left(\frac{I(p)-m_2}{m_2-m_1}\right)^2 & \tau < I(p) \leq m_2 \\ 2 & I(p) > m_2 \end{cases} \tag{3.1}$$

where τ is the Otsu's threshold, and m_1 and m_2 are the mean intensities of each cluster. We compute a 3D gradient at each voxel as the sum of its projections along 26 directions around the voxel, and then use its magnitude for tree pruning (Figure 3.1c).

Seed Selection. The brighter cluster contains many voxels outside the brain (Figure 3.1b). To obtain a set of seeds inside the brain, we apply a morphological erosion by S_5 on the binary image of the brighter cluster. This operation disconnects the brain from adjacent structures. We then select the largest connected component as the seed set (Figure 3.1d).

Morphological Closing. The brain object obtained by tree pruning (Figure 3.1e) might not include the entire longitudinal fissure, especially when the fissure is too thick. To ensure its inclusion, we apply a morphological closing by S_{20} to the binary brain image (Figure 3.1f).

Thick CSF Structure Removal. The last step of this phase is the removal of thick CSF structures (such as the ventricles, lesions and post-surgery cavities) from the brain object, to avoid the MSP from snapping to a dark structure other than the longitudinal fissure. We achieve this with a sequence of morphological operations: we start from a binary image obtained by thresholding at Otsu's optimal threshold (Figure 3.1b). We apply a morphological opening by S_5 to connect the thick (> 5 mm) CSF structures (Figure 3.1g), and then dilate the result by S_2 to include a thin (2 mm) wall of the CSF structures (Figure 3.1h). This dilation ensures the reinclusion of the longitudinal fissure, in case it is removed by the opening. The binary intersection of this image with the brain object is then used as brain mask (Figure 3.1i) by the next stage of our method. Only voxels within this mask are considered by stage 2. Figures 3.2a and 3.2b show how the computed brain mask excludes the large cavity in a post-surgery image, and figures 3.2c and 3.2d show how the mask excludes most of the ventricles in patients with large ventricles.

3.3.2 Plane Location Stage

To obtain the CSF score of a plane, we compute the mean voxel intensity in the intersection between the plane and the brain mask (Figures 3.3a and 3.3b). The lower the score, the more likely the plane is to contain more CSF than white matter and gray matter. The plane with a sufficiently large brain mask intersection and minimal score is the most likely to be the mid-sagittal plane.

To find a starting candidate plane, we compute the score of all sagittal planes in 1 mm intervals (which leads to 140–180 planes in usual MR datasets), and select the plane with minimum score. Planes with intersection area lower than 10 000 mm^2 are not considered to avoid selecting planes tangent to the surface of the brain. Planes with small intersection areas may lead to low scores due to alignment with sulci and also due to partial volume effect between gray matter and CSF (Figures 3.3c and 3.3d).

Once the best candidate plane is found, we compute the CSF score for small transformations of the plane by a set of rotations and translations. If none of the transformations lead to a plane with lower CSF score, the current plane is the mid-sagittal plane and the algorithm stops. Otherwise, the transformed plane with lower CSF score is considered the current candidate, and the algorithm is repeated. The algorithm is finite, since each iteration reduces the CSF score, and the CSF score is limited by the voxel intensity domain.

We use a set of 42 candidate transforms at each iteration: translations on both directions of the X, Y and Z axes by 10 mm, 5 mm and 1 mm (18 translations) and rotations on both directions around the X, Y and Z axes by 10º, 5º, 1º and 0.5º (24 rotations). All rotations are about the central point of the initial candidate plane. There is no point in attempting rotations by less than 0.5º, as this is close to the limit where planes fall over the same voxels for typical MR datasets, as discussed in Section 3.4.1.

3.4 Evaluation and Discussion

3.4.1 Error Measurement

The discretization of $\mathbb{R}^3$ makes planes that differ by small angles to fall over the same voxels. Consider two planes A and B that differ by an angle Θ (Figure 3.4). The minimum angle that makes A and B differ by at least 1 voxel at a distance r from the rotation center is given by Equation 3.2.

$$\Theta = \arctan\left(\frac{1}{r}\right) \tag{3.2}$$

An MR dataset with 1 mm^3 voxels has a typical maximum dimension of 256 mm. For rotations about the center of the volume, the minimum angle that makes planes A and B differ by at least one voxel within the volume (point p_i in Figure 3.4) is approximately $\arctan\left(\frac{1}{128}\right) = 0.45^{\text{o}}$. For most MSP applications, we are only concerned about plane differences within the brain. The largest length within the brain is usually longitudinal, reaching up to 200 mm in adult brains. The minimum angle that makes planes A and B differ by at least one voxel within the brain (point p_b in Figure 3.4) is approximately $\arctan\left(\frac{1}{100}\right) = 0.57^{\text{o}}$.

Therefore, we can consider errors around 1º excellent and equivalent results.

3.4.2 Experiments

We evaluated the method on 64 MR datasets divided in 3 groups: A control group with 20 datasets from subjects with no anomalies, a surgery group with 36 datasets from patients with significant structural variations due to brain surgery, and a phantom group with 8 synthetic datasets with varying levels of noise and inomogeneity, taken from the BrainWeb project [21].

All datasets in the control group and most datasets in the surgery group were acquired with a voxel size of $0.98 \times 0.98 \times 1.00$ mm^3. Some images in the surgery group were acquired with a voxel size of $0.98 \times 0.98 \times 1.50$ mm^3. The images in the phantom group were generated with an isotropic voxel size of 1.00 mm^3. All volumes in the control and surgery groups were interpolated to an isotropic voxel size of 0.98 mm^3 before applying the method.

For each of the 64 datasets, we generated 10 variations (tilted datasets) by applying 10 random transforms composed of translations and rotations of up to 12 mm and 12^o in all axes. The method was applied to the 704 datasets (64 untilted, 640 tilted), and visual inspection showed that the method correctly found acceptable approximations of the MSP in all of them. Figure 3.5 shows sample slices of some datasets and the computed MSPs.

For each tilted dataset, we applied the inverse transform to the computed mid-sagittal plane to project it on its respective untilted dataset space. Thus, for each untilted dataset we obtained 11 planes which should be similar. We measured the angle between all $\binom{11}{2} = 55$ distinct plane pairs. Table 3.2 shows the mean and standard deviation (σ) of these angles within each group. The low mean angles (column 3) and low standard deviations (column 4) show that the method is robust with regard to linear transformations of the input. The similar values obtained for the 3 groups indicate that the method performs equally well on healthy, pathological and synthetic data. The majority (94.9%) of the angles were less than 3^o, as shown in the histogram of Figure 3.6. Of $64 \times 55 = 3520$ computed angles, only 5 (0.1%) were above 6^o. The maximum measured angle was 6.9^o. Even in this case (Figure 3.7), both planes are acceptable in visual inspection, and the large angle between different two computations of the MSP can be related to the non-planarity of the fissure, which allows different planes to match with similar optimal scores. The lower mean angle in the phantom group (column 3, line 3 of Table 3.2) can be related to the absence of curved fissures in the synthetic datasets. Figure 3.8 shows some examples of non-planar fissures.

Tabela 3.2: Angles between computed MSPs

Group	**Datasets**	**Angles**	
		Mean	σ
Control	20	1.33^o	0.85^o
Surgery	36	1.32^o	1.03^o
Phantom	8	0.85^o	0.69^o
Overall	64	1.26^o	0.95^o

All experiments were performed on a 2.0 GHz Athlon64 PC running Linux. The method took from 41 to 78 seconds to compute the MSP on each MR dataset (mean: 60.0 seconds). Most of the time was consumed computing the brain mask (stage 1).

Stage 1 required from 39 to 69 seconds per dataset (mean: 54.8 seconds), while stage 2 required from 1.4 to 20 seconds (mean: 5.3 seconds). The number of iterations in stage 2 ranged from 0 to 30 (mean: 7.16 iterations).

3.5 Conclusions and Future Work

We presented a fast and robust method for extraction of the mid-sagittal plane from MR images of the brain. It is based on automatic segmentation of the brain and on a heuristic search based on maximization of CSF within the MSP. We evaluated the method on 64 MR datasets, including images from patients with large surgical cavities (Figure 3.2a and Figures 3.5e–h). The method succeeded on all datasets and performed equally well on healthy and pathological cases. Rotations and translations of the datasets led to mean MSP variations around 1º, which is not a significant error considering the discrete space of MR datasets. MSP variations over 3º occurred only in cases where the longitudinal fissure was not planar, and multiple planes fitted different segments of the fissure with similar scores. The method required a mean time of 60 seconds to extract the MSP from each MR dataset on a common PC.

Previous fissure-based works were either evaluated on images of healthy patients, on images with small lesions [98], or relied on local symmetry measurements [56]. As future work, we intend to implement some of the previous works and compare their accuracy and performance with our method on the same datasets. Brain mask computation is responsible for most of the computing time. We also plan to evaluate how the computation of the brain mask on lower resolutions affect the accuracy and efficiency of the method.

ACKNOWLEDGEMENTS

The authors thank CAPES (Proc. 01P-05866/2007), CNPq (Proc. 302427/04-0), and FAPESP (Proc. 03/13424-1) for the financial support.

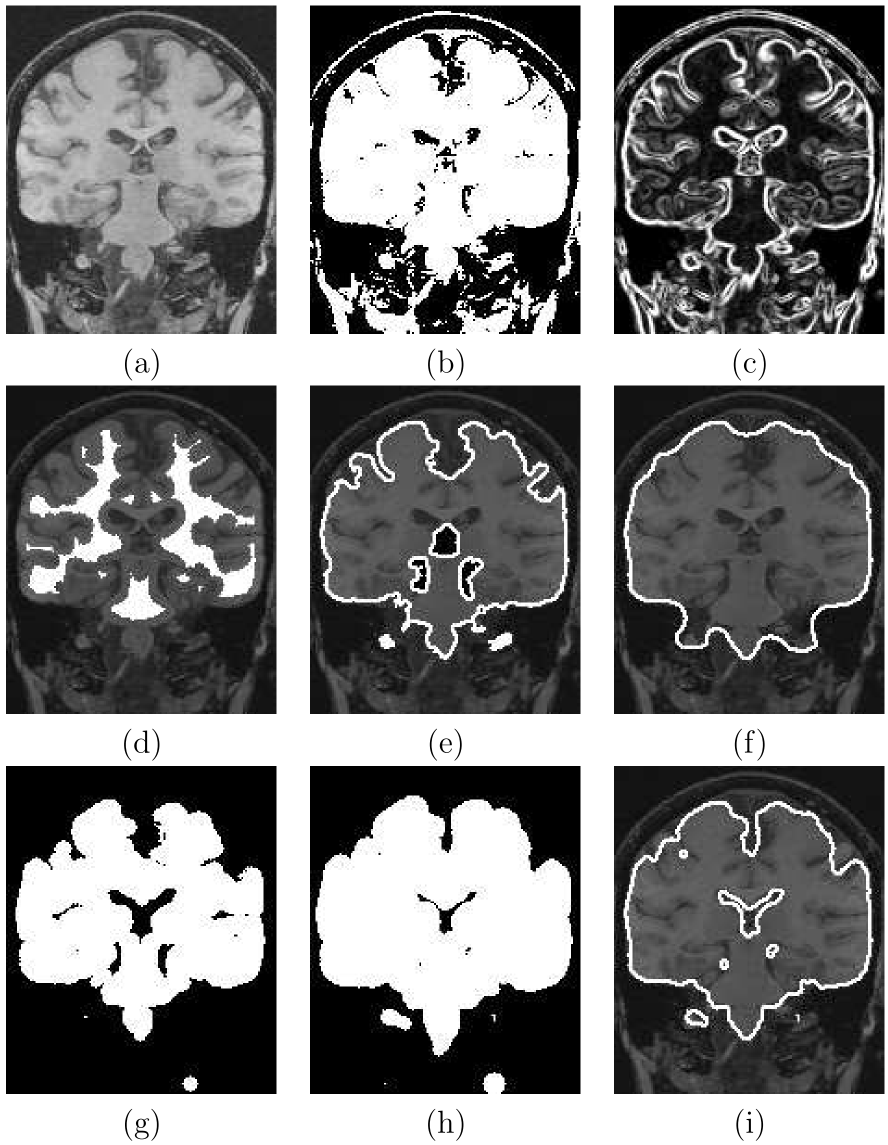
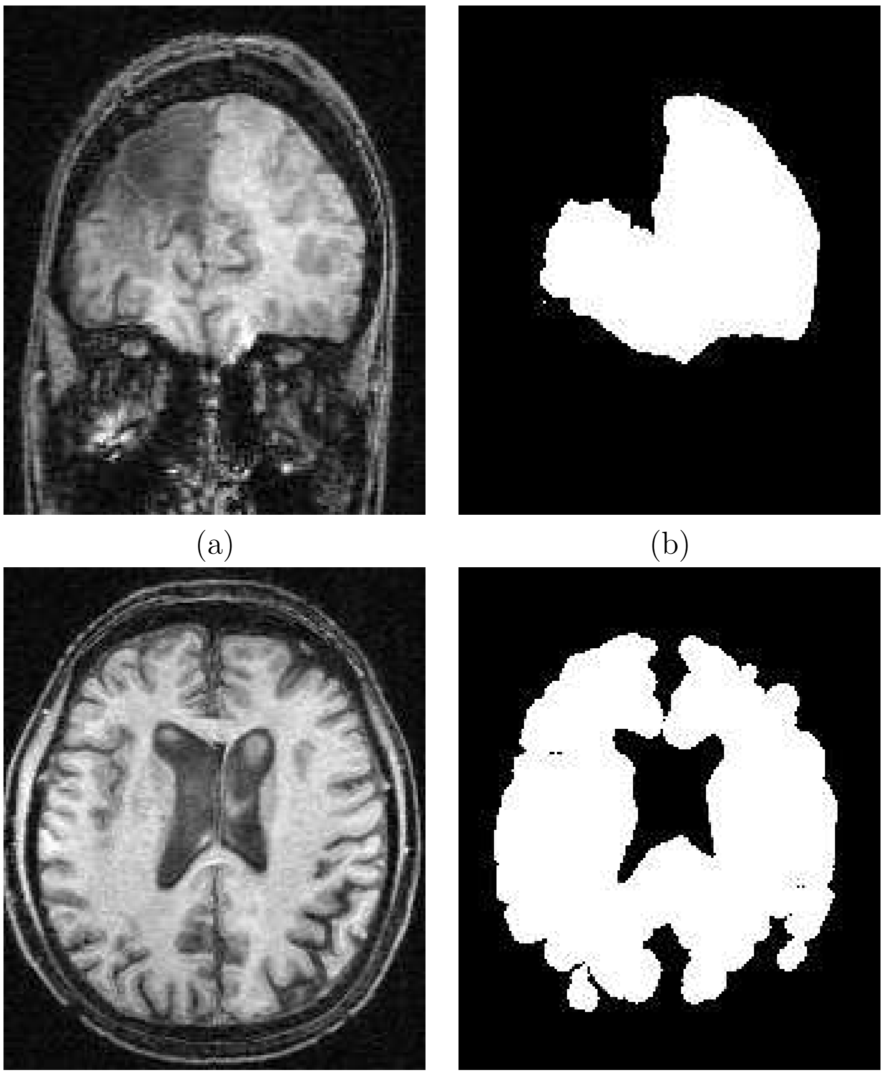


Figura 3.1: Sample slice of the intermediary steps in stage 1: (a) original coronal MR slice; (b) binary cluster mask obtained by thresholding; (c) gradient-like image used for tree pruning; (d) seed set used for tree pruning (white); (e) border of the brain object obtained by tree pruning (white); (f) border of the brain object after morphological closing; (g) CSF mask after opening; (h) CSF mask after dilation; (h) brain mask (intersection of (f) and (h)).



(a) (b)

(c) (d)

Figura 3.2: Examples of thick CSF structure removal: (a) coronal MR slice of a patient with post-surgical cavity; (b) brain mask of (a); (c) axial MR slice of a patient with large ventricles; (d) brain mask of (c).

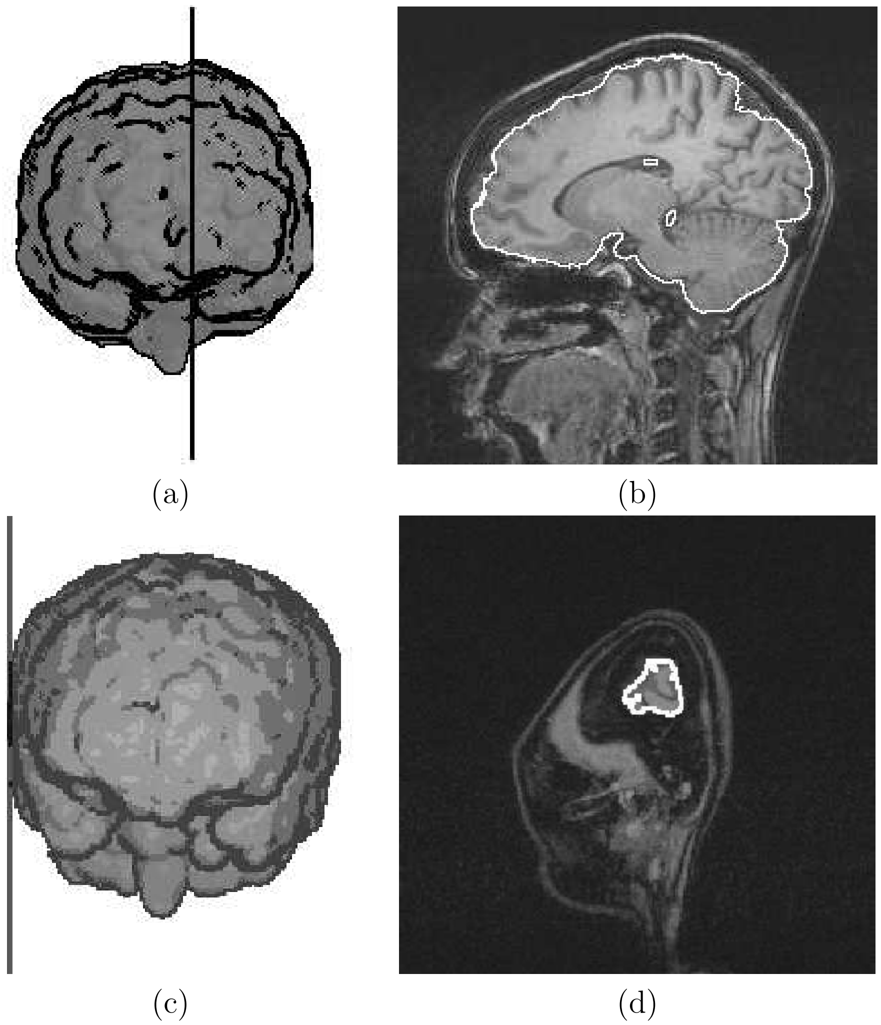


Figura 3.3: Plane intersection: (a–b) sample plane, brain mask and their intersection (white outline). (c–d) example of a plane tangent to the brain's surface and its small intersection area with the brain mask (delineated in white), overlaid on the original MR image.

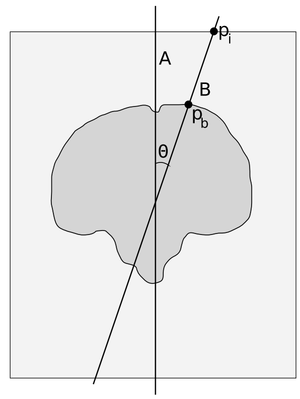


Figura 3.4: Error measurement in discrete space: points and angles.

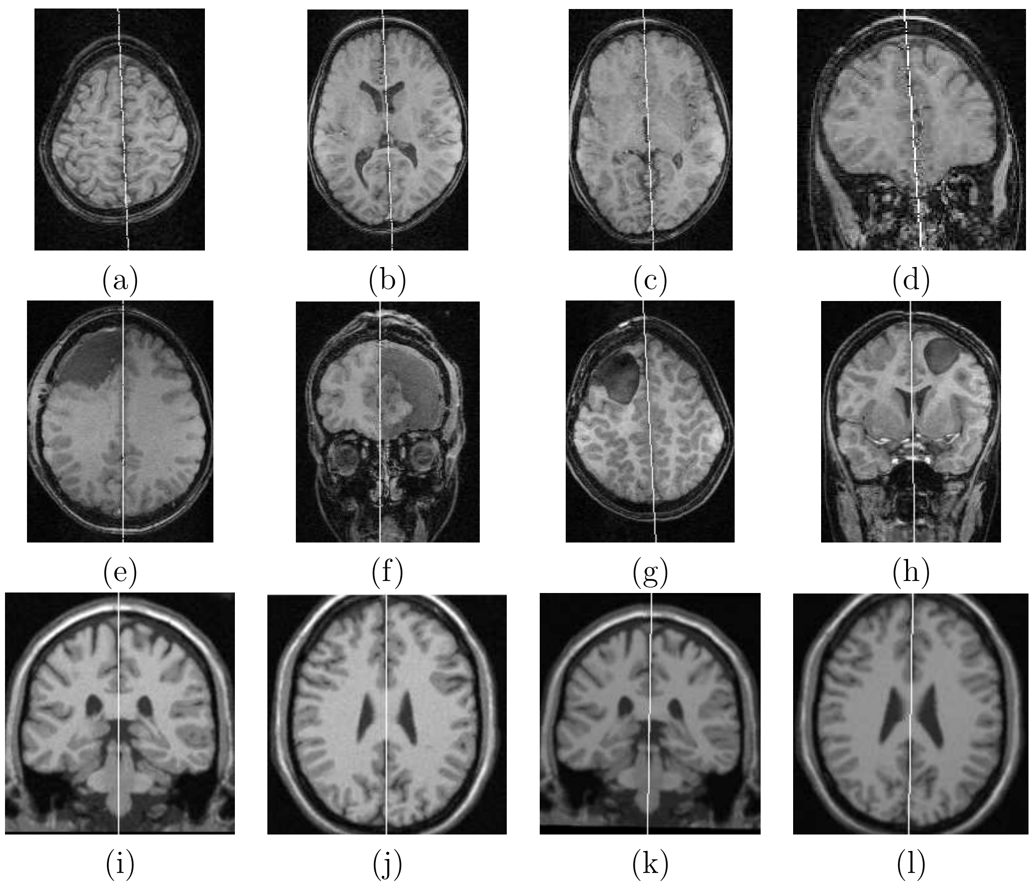


Figura 3.5: Examples of planes computed by the method: (a–d): sample slices from a control dataset; (e–f) sample slices from a surgery dataset; (g–h) sample slices from another surgery dataset; (i–j): sample slices from a phantom dataset; (k–l): sample slices from a tilted dataset obtained from the one in (i–j).

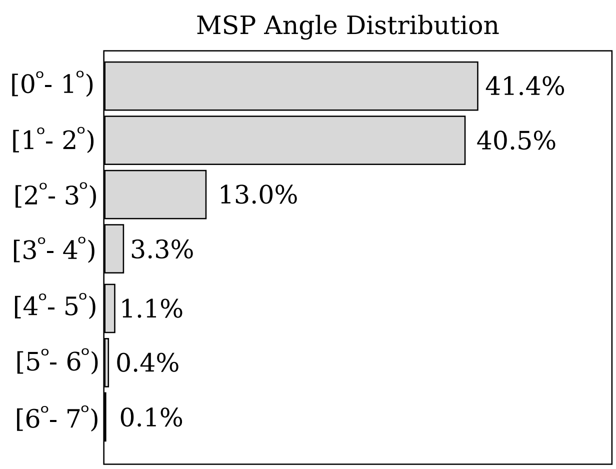
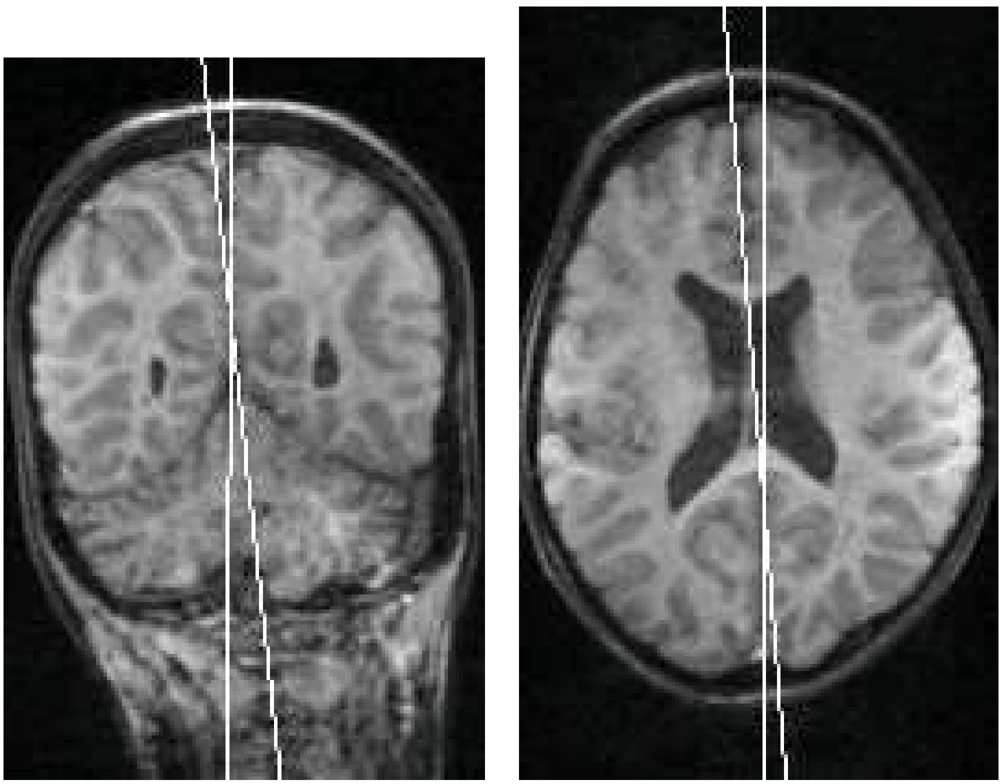


Figura 3.6: Distribution of the angles between computed mid-sagittal planes.



(a) (b)

Figura 3.7: A coronal slice (a) and an axial slice (b) from the case with maximum angular error (6.9º), with planes in white: The fissure was thick at the top of the head, and curved in the longitudinal direction, allowing the MSP to snap either to the frontal or posterior segments of the fissure, with some degree of freedom.

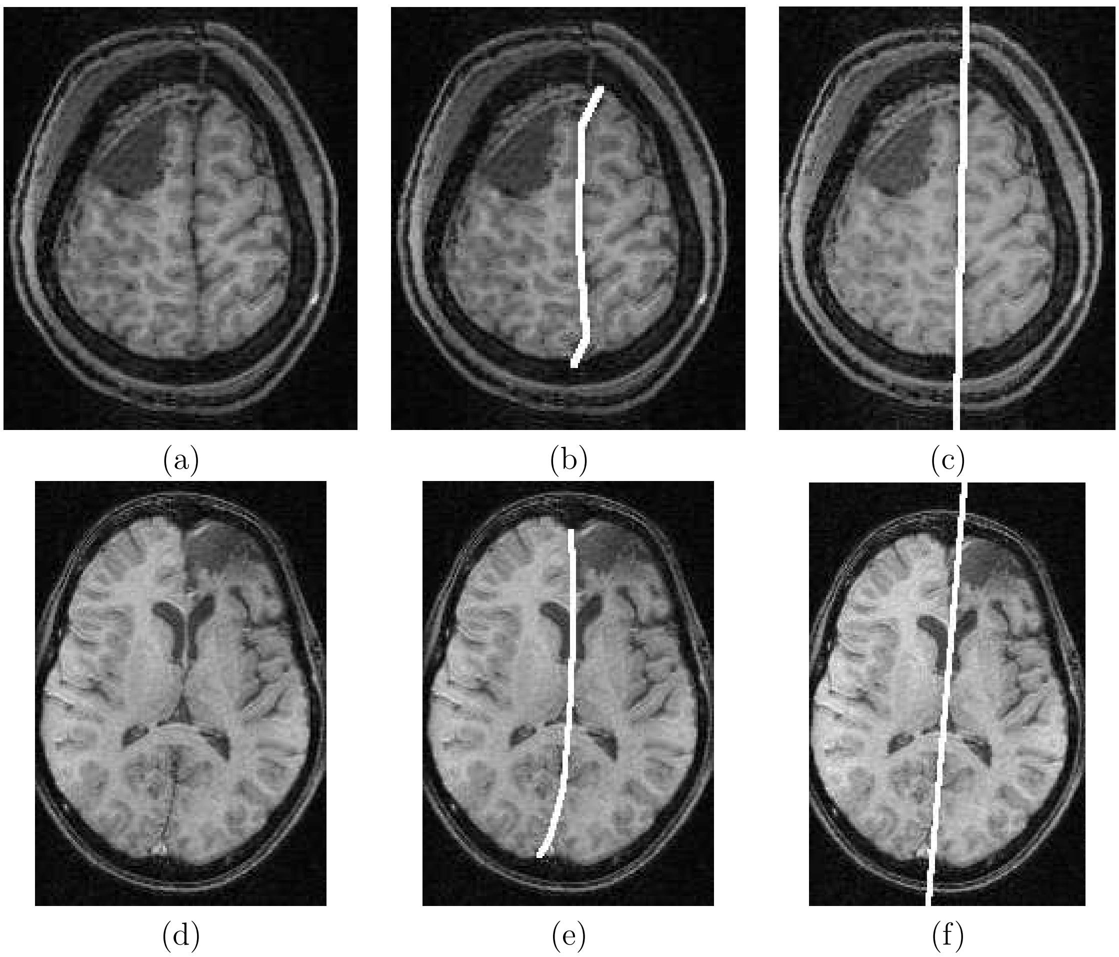


Figura 3.8: Non-planar fissures: (a) irregular fissure, (b) expert fissure delineation of (a) and (c) MSP computed by our method. (d) Curved fissure, (e) expert fissure delineation of (d) and (f) MSP computed by our method.

Capítulo 4

Fast and Automatic Curvilinear Reformatting of MR Images of the Brain for Diagnosis of Dysplastic Lesions

4.1 Introduction

The diagnosis of dysplastic lesions in the human brain is an important task to guide the treatment of refractory epilepsy patients, which often requires surgical removal of the lesion's tissue. While a dysplastic lesion can only be confirmed by histological analysis (which requires invasive procedures), its location and extent appear in MR images of the brain as blurrings of voxel intensity with gray-matter texture. It is known that curvilinear reformatting is the best non-invasive technique for diagnosis of dysplastic lesions [5, 7, 24]. It consists of computing surfaces that follow the brain's curvature at various depths, such that the diagnosis is possible by visual inspection of the voxel intensities on these surfaces. This technique stems from the fact that the lesions are enhanced on surfaces which are orthogonal to most of the sulci. However, the unique implementation of curvilinear reformatting requires user intervention [79]. The computation of the surfaces relies on manual delineation of lines on a few 2D slices followed by surface interpolation— a task that is prone to user mistakes and leads to curvature artifacts.

We present a fast method for curvilinear reformatting that solves both problems. Our method uses a general graph-based approach to segment the brain, extract its *envelope* (a smooth surface that follows its curvature), and compute the isosurfaces at all possible depths by euclidean distance transform. It requires no user input, no *ad-hoc* parameters, and takes less than 1 minute to run on a common PC.

We could have used any other method for automatic brain segmentation, but the method proposed here is a recent and promising contribution under evaluation. It does not require *ad-hoc* parameters or templates, unlike most known approaches [47, 19].

4.2 Brain Segmentation

A possible envelope for curvilinear reformatting is the surface of the dura-matter. The dura-mater is a thin membrane and its automatic segmentation is a complicated issue. We take an easier approach which is to segment the brain— gray matter (GM) and white matter (WM)— and then use the surface of its morphological closing [35] as envelope. Our method is entirely based on the Image Foresting Transform (IFT) for fast brain segmentation, morphological operations, and euclidean distance transform of the envelope [39].

The IFT interprets an MR image as a graph whose nodes are the voxels and whose arcs are defined by an *adjacency relation* between voxels. For a given set of seed voxels and a suitable *path-cost function*, the IFT reduces many image processing problems to the computation of an optimum-path forest in this graph, such that each tree consists of voxels more strongly connected to its root than to any other seed in some appropriate sense. This approach has also the advantage of being computable in time proportional to the number of voxels in most applications.

We use an IFT-based operator, called *tree pruning* [37, 12], which takes as input a gradient-like image, seed voxels inside the brain and 6-neighborhood adjacency relation. The path-cost function in tree pruning is a mapping that assigns to every path in the graph the maximum gradient intensity along that path. In these conditions, the IFT leads to a forest where brain and background are connected by a few optimum paths. These paths cross the brain's boundary through its "most weakly connected" parts (bottleneck voxels). The topology of the forest is exploited to identify the bottleneck voxels and prune their subtrees, such that the remaining forest defines the brain.

The gradient-like image must be brighter on the brain's boundary than inside it and in the nearby background. This gradient condition is paramount for the success of the method. For further details about tree pruning, see [37, 12].

In the next sections we show how to obtain a suitable gradient-like image and the seeds inside the brain.

4.2.1 Gradient Estimation

MR-T1 images of the brain contain two large clusters: the first with air, bone and cerebrospinal fluid (CSF), represented by darker voxels, and the second, represented by brighter voxels, consists of GM, WM, skin, fat and muscles. Otsu's optimal threshold [74]

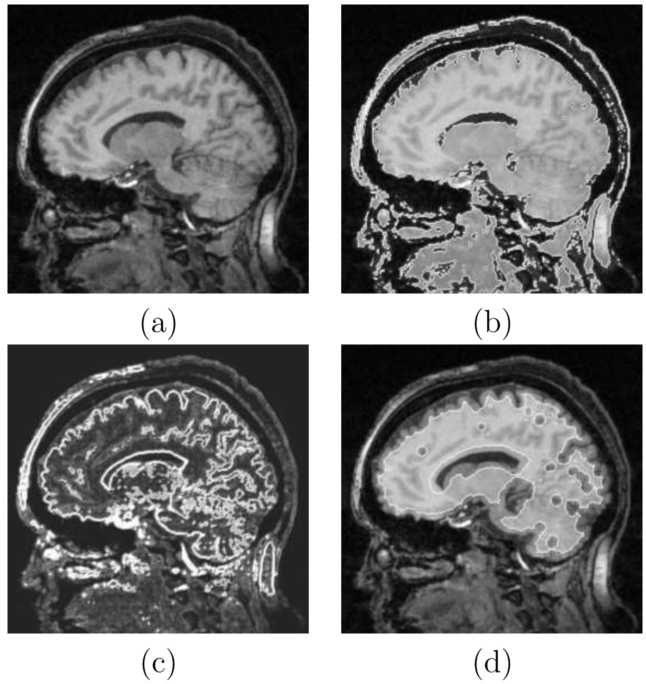


Figura 4.1: (a) Sample slice of MR-T1 image of the brain. (b) Binary image resulting from Otsu's thresholding overlaid on image (a). (c) Gradient-like image. (d) Binary image of seed voxels overlaid on image (a).

can separate these clusters (Figs. 4.1a and 4.1b), such that the GM/CSF border becomes part of the border between them. To enhance the GM/CSF border in the original image, we multiply each voxel intensity $I(p)$ by a weight $w(p)$ as follows:

$$w(p) = \begin{cases} 0 & I(p) \leq m_1 \\ 2\left(\frac{I(p)-m_1}{m_2-m_1}\right)^2 & m_1 < I(p) \leq \tau \\ 2-2\left(\frac{I(p)-m_2}{m_2-m_1}\right)^2 & \tau < I(p) \leq m_2 \\ 2 & I(p) > m_2 \end{cases} \tag{4.1}$$

where τ is the Otsu's threshold, and m_1 and m_2 are the mean intensities computed on each cluster. We then compute the 3D morphological gradient [35] with a 6-neighborhood structuring element (Fig. 4.1c).

4.2.2 Seed selection

The brighter cluster contains many voxels outside the brain (Fig. 4.1b). To obtain a set of seeds inside the brain, we apply a morphological erosion by a spherical structuring element of radius 5 on the binary image of the brighter cluster. This operation disconnects the brain from neighboring structures. We then select the largest connected component as the seed set (Fig. 4.1d).

The choice of the structuring element's radius for erosion depends on the voxel size. The value presented here was adjusted for images with an isotropic voxel size of 1 mm. This value guarantees the disconnection of the brain from the background and a reasonable number of seeds inside the brain. It is important to note that automatic brain segmentation cannot be achieved with only morphological operations.

4.3 3D visualization of isosurfaces

We compute the morphological closing of the brain by a spherical structuring element of radius 20. The resulting surface is used as envelope. This radius also depends on the voxel size, as discussed above.

4.3.1 Distance map computation

The euclidean distance map of the envelope is computed by an IFT, where the seeds are its voxels, the adjacency relation is the 26-neighborhood (a spherical relation of radius $\sqrt{3}$), and the path-cost function is a mapping that assigns to each path in the graph the euclidean distance between its endpoints. Up to this point, the process is fully automatic. The distance map of the envelope effectively encodes all isosurfaces.

4.3.2 Isosurface visualization

Once the user selects a viewing direction and surface depth d (distance from the envelope) for inspection, the isosurface can be rendered by voxel splatting, projecting only voxels within a distance $d + \epsilon$, for a small ϵ, from the envelope. Projection is sped up by sorting the voxels by distance. The small delay for sorting the voxels after the distance transform allows faster renderizations, since the first voxel to be projected can be efficiently located by binary search in the sorted array of voxels, avoiding a raster scan of the entire image.

4.4 Evaluation

To evaluate our method, we selected 5 MR images from refractory epilepsy patients (all of them had dysplastic lesions confirmed by histological analysis performed after image acquisition) and 10 MR images from control subjects with no known anomalies. All images were acquired in an Elscint 2T MR scanner with a voxel size of $0.98 \times 0.98 \times 1.00$ mm, and were interpolated to an isotropic voxel size of 1.00 mm before all further processing.

4.4.1 Segmentation

The segmentation method succeeded on all 15 subjects. Fig. 4.2 shows 3D renderizations of the segmented brains obtained with automatic tree pruning (top row) and their respective envelopes (bottom row) for 2 control subjects and 2 patients.

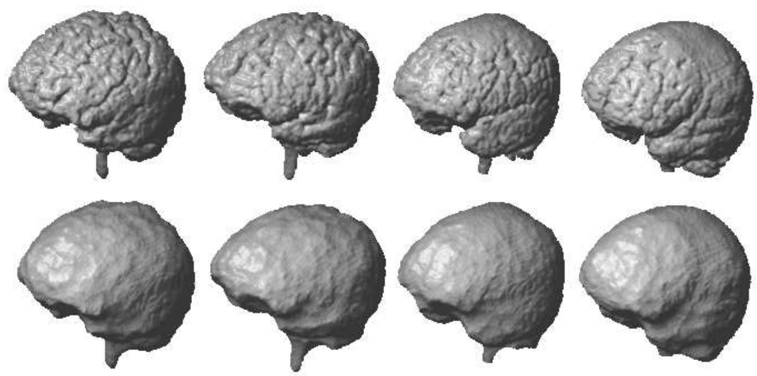


Figura 4.2: 3D renderizations of the brain segmentations obtained with the automatic tree pruning approach (top row) and of the resulting envelopes (bottom row). The leftmost two columns are from control subjects and the other two columns are from epilepsy patients.

An Athlon64 3200+ PC took an average time of 38 seconds to perform the brain segmentation and envelope computation (8.5 seconds for Otsu threshold computation, gradient estimation and seed set computation, 17.1 seconds for automatic tree pruning and 12.4 seconds for the envelope computation, on average). Image sizes varied from 9.3×10^6 to 11.7×10^6 voxels.

4.4.2 Isosurface Visualization

Fig. 4.3 shows sample isosurface renderizations obtained with our method, and Fig. 4.4 shows dysplastic lesions on 4 patients. Lesions are usually detected by locating texture assymetries between the brain hemispheres.

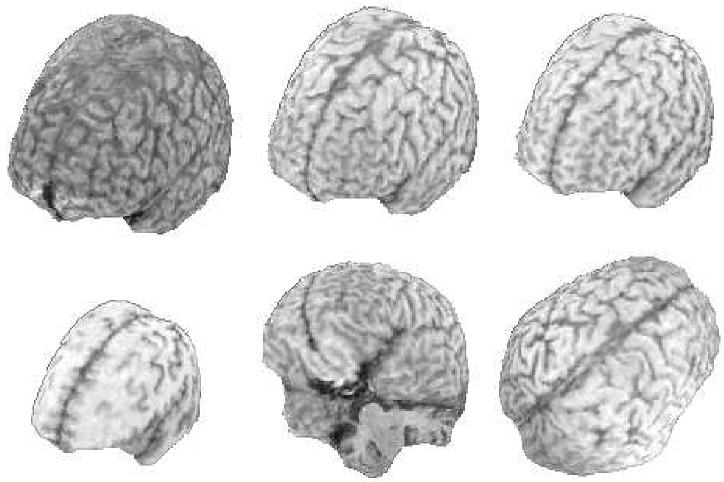


Figura 4.3: Isosurface renderizations of the brain of an epilepsy patient, at various depths and viewing directions.

On the same Athlon64 3200+ PC, the euclidean distance transform took 5 seconds to be computed for each image, on average. The distance map sorting took an average time of 2 seconds. Our evaluation implementation uses software-based voxel splatting, which achieves rates between 5 and 15 frames per second, depending on surface depth and zoom level. This rate is enough to provide interactivity and a responsive diagnostic tool. It could be further improved by exploiting the hardware acceleration of modern GPUs.

4.4.3 Comparison to Other Methods

For the segmentation task, we attempted to use the widely available and well-known SPM2 [47], but it made so many segmentation mistakes outside the brain that the results were unusable to obtain an envelope without considerable additional effort (Fig. 4.5). The SPM2 segmentation can be improved with a better template and some image preprocessing, but it is still much slower than our method. SPM2 took from 5 to 7 minutes to segment each image, while automatic tree pruning took 38 seconds.

For isosurface visualization, the existing approach is the BrainSight software tool [79], which requires manual delineation of lines following the brain's curvature on a few 2D slices (Figs. 4.6a and 4.6b). The surface used as reference for curvilinear reformatting is obtained by interpolation (Fig. 4.6c). Note that the frontal and occipital lobes suffer from curvature artifacts and the resulting visualization is deformed on the "end caps".

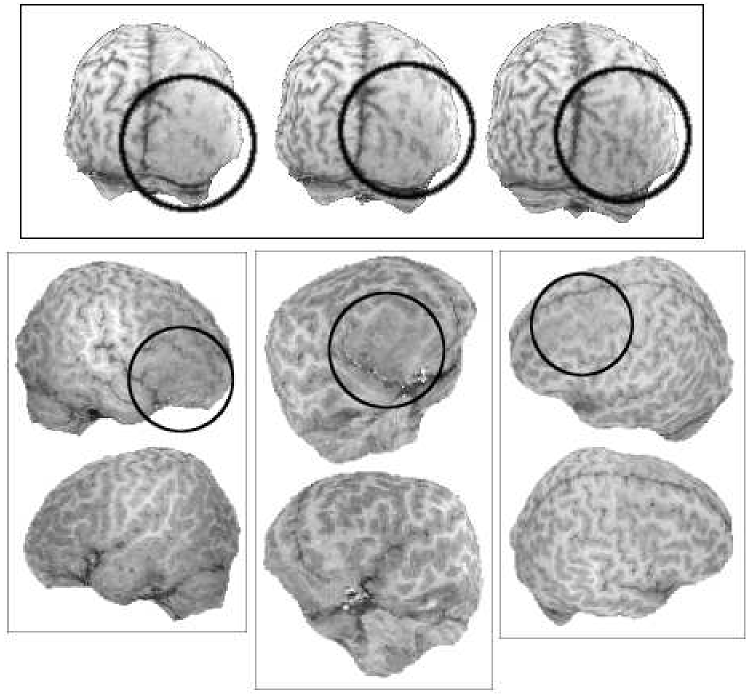


Figura 4.4: Dysplastic lesions on four patients. The top box with three views is the lesion on the patient from Fig. 4.3 viewed at different depths. On the other 3 patients, the opposing side is shown for texture assymetry comparison. All lesions are indicated by a black ring.

4.5 Conclusions and Future Work

We presented a fast and fully automatic approach for curvilinear reformatting of MR images of the brain based on the image foresting transform [39, 37]. The technique is useful for the non-invasive diagnosis of dysplastic lesions in the human brain. These lesions are a major cause of refractory epilepsy and their diagnosis is crucial for treatment planning [5, 7, 24].

The proposed solution consists of an automatic image processing pipeline that takes raw MR images and outputs 3D visualization of isosurfaces of the brain's envelope in less than 1 minute on a common PC. Our method does not present curvature artifacts such as the curvilinear reformatting approach currently available [7, 79]. Our segmentation step is also much faster than widely used approaches, such as SPM2 [47].

We are currently evaluating our automatic brain segmentation method using a larger number of data sets, and future work includes the implementation of a hardware-

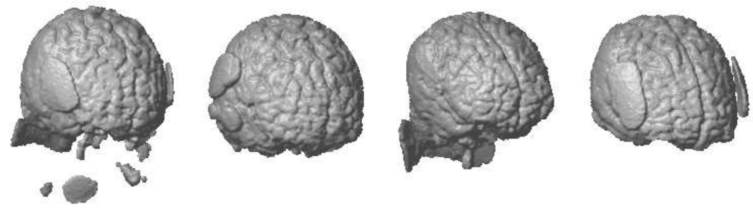


Figura 4.5: 3D renderizations of incorrect brain segmentations obtained with SPM2.

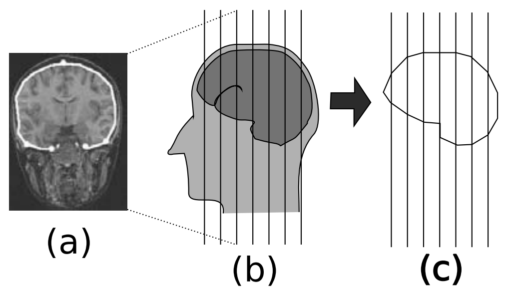


Figura 4.6: Curvilinear reformatting in BrainSight. (a) Manual delineation in a coronal slice. (b) A few other coronal slices where manual delineation is performed. (c) The interpolated surface used as reference for curvilinear reformatting.

accelerated isosurface visualization software tool.

Capítulo 5

FCD Segmentation using Texture Asymmetry of MR-T1 Images of the Brain

5.1 Introduction

Focal Cortical Dysplasia (FCD) is a malformation of cortical development that results in abnormal glial elements and disruption of the normal cortical lamination. It was first described by Taylor [90], and it is the most common malformation in patients with intractable epilepsy [48]. FCD is also a common cause of epilepsy in children and young adults [24, 81].

The problem of FCD detection in MRI consists in identifying the approximate locations of the FCD lesions (usually one voxel inside each lesion, or an approximation of the lesion by some kind of marker). The problem of FCD segmentation consists in identifying the precise spatial extent of FCD lesions, by classifying MRI voxels as either healthy or pathological. Solving FCD segmentation implicitly solves FCD detection. Detection and segmentation of FCD lesions are crucial steps in treatment planning [75, 84, 64].

FCD lesions appear as subtle macroscopic features in brain MR images: blurring of the gray-matter/white-matter (GM/WM) transition, localized GM thickening and hyperintense GM signal [24]. In this work we exploit the asymmetry between texture features of lesional voxels and their healthy counterparts in the opposing brain hemisphere to detect and segment FCD lesions.

5.2 Related Works

The most common FCD diagnosis method in clinical practice is the straightforward visual inspection of MR images by specialists. The reported detection rate for this technique is 50% [72]. Techniques such as multiplanar reconstruction (MPR) and curvilinear reformatting (CR) increase the detection rate to 100% [72]. Curvilinear reformatting [7] consists of visualizing 3D MR images as curved surfaces that follow the shape of the brain. This technique improves the visibility of FCD lesions over MPR [72] and can be computed automatically without human interaction [10].

Several recent works presented automatic FCD detection methods, with detection rates varying from 53% to 85% [60, 2, 86, 23]. Most of these works focus on detection rather than segmentation. To our knowledge, Colliot et al. [23] is the only work that report segmentation accuracy rate, with a coverage of 73% of the lesional voxels. Some of these works [60, 2, 23] rely on template-based segmentation techniques of the GM, which are often unreliable for children and patients who underwent brain surgery [21, 30].

5.3 Method

Our method works on volumetric MR-T1 images interpolated to an isotropic voxel size of $1.0\,mm^3$, and comprises 6 steps: (i) mid-sagittal plane (MSP) location, (ii) brain segmentation, (iii) CR computation, (iv) feature extraction, (v) voxel classification and (vi) outlier removal.

5.3.1 MSP Location

We locate the mid-sagittal plane that divides the brain hemispheres using the heuristic minimization search of Bergo et al. [14]. After the plane is found, the volume is rotated such that the MSP becomes orthogonal to the z axis.

5.3.2 Brain Segmentation

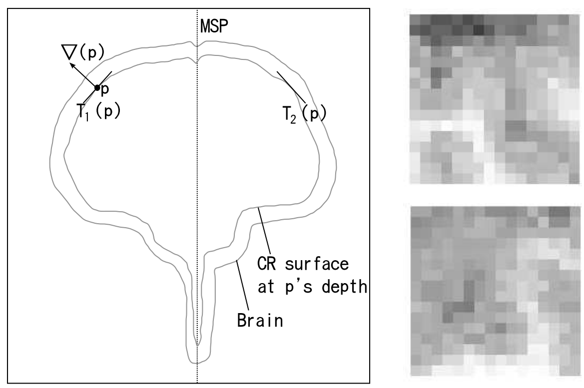
We apply the automatic tree pruning technique [12] to segment the brain. Tree pruning does not rely on templates and performs well regardless of age or anatomic variations [14].

5.3.3 CR computation

The curvilinear reformatting can be encoded as an Euclidian distance transform computed from the brain's border. This distance transform can be efficiently computed by the IFT-EDT [10].

5.3.4 Feature Extraction

For each voxel p within the brain, we extract a 16×16 planar texture patch $T_1(p)$ tangent to the brain's curvature (as computed by the CR) and centered at p. The gradient vector of the CR distance transform at the voxel's location provides the surface normal. We also extract a symmetric patch $T_2(p)$, located at the reflection of $T_1(p)$ by the MSP. These patches are illustrated by Fig. 5.1.



(a) (b)

Figura 5.1: Texture patches used for feature extraction: (a) patch location. (b) example of a pair of symmetric patches $T1$ and $T2$.

The patch size was chosen experimentally. Smaller patch sizes did not provide good classification results, while larger patch sizes led to similar results with higher computational cost.

For each patch we compute 6 features: sharpness (h), entropy, homogeneity, contrast, intensity mean (μ) and intensity standard deviation (σ). Sharpness is computed as the sum of pixelwise absolute intensity differences between the patch and a blurred copy of itself obtained by convolution with a 5×5 Gaussian kernel with $\sigma = 7$. Entropy, homogeneity and contrast are computed as presented by Haralick et al. [52], using a 12×12 gray-level cooccurrence matrix. Using these computed values we build a 16-element feature vector associated to voxel p as indicated by Eq. 5.1. All features are scaled to fit within the $[0, 1]$ interval.

$$fv(p) = \begin{bmatrix} h(T_1(p)) \\ h(T_2(p)) \\ h(T_1(p)) - h(T_2(p)) \\ h(T_1(p))/h(T_2(p)) \\ entropy(T_1(p)) \\ entropy(T_2(p)) \\ contrast(T_1(p)) \\ contrast(T_2(p)) \\ homogeneity(T_1(p)) \\ homogeneity(T_2(p)) \\ \mu(T_1(p)) \\ \mu(T_2(p)) \\ \mu(T_1(p)) - \mu(T_2(p)) \\ \sigma(T_1(p)) \\ \sigma(T_2(p)) \\ \sigma(T_1(p)) - \sigma(T_2(p)) \end{bmatrix} \tag{5.1}$$

5.3.5 Voxel Classification

We use a Reduced Coulomb Energy classifier (RCE) [36] to perform the classification of the voxels, based on supervised learning. In the RCE classifier, each training sample becomes a hypersphere in the feature space (in our case, $\mathbb{R}^{16}$). The radius is chosen to be the maximum such that no training sample from a different class is contained in the hypersphere. Classification is performed by testing the test sample for containment within the training samples's hyperspheres. If a test sample falls within an ambiguous space (being contained by hyperspheres of different classes), we classify it as lesional. This is a design decision to prevent false negatives.

5.3.6 Outlier Removal

The classification result leads to outliers both in lesional and healthy regions. However, the density of voxels classified as lesional is visibly higher within the actual lesions. In this step we threshold the density of voxels classified as lesional within a fixed adjacency radius, and consider lesional only those voxels with a density above a certain threshold, determined experimentally.

5.4 Results

We used images from 5 epilepsy patients (age/gender: 11/M, 24/F, 12/M, 51/M, 30/F) with confirmed FCDs to evaluate the method. The lesions were manually segmented by an specialist with knowledge about the lesion locations, and this ground truth was used to extract a small training set of feature vectors from each patient. We used 1% of the voxels from each patient to build the training set, preserving a ratio of 1 : 9 between lesional and healthy voxels. This ratio affects the radii of the RCE hyperspheres and can be used to control the ratio of false positives/false negatives in any direction.

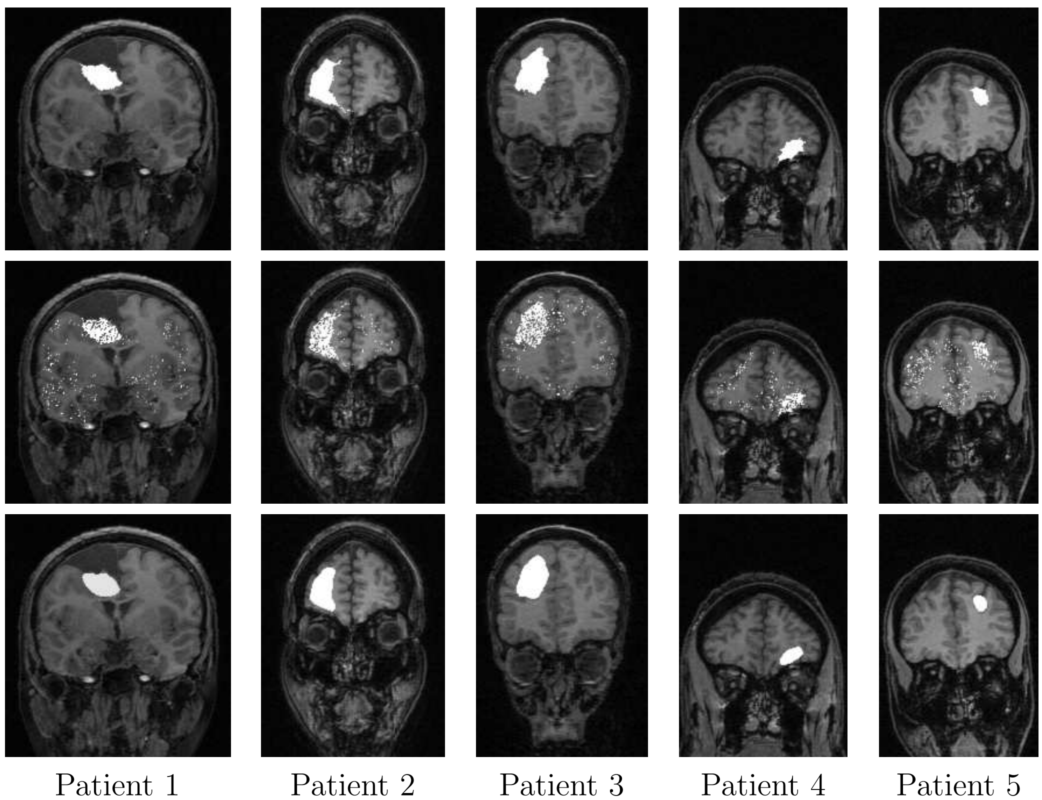
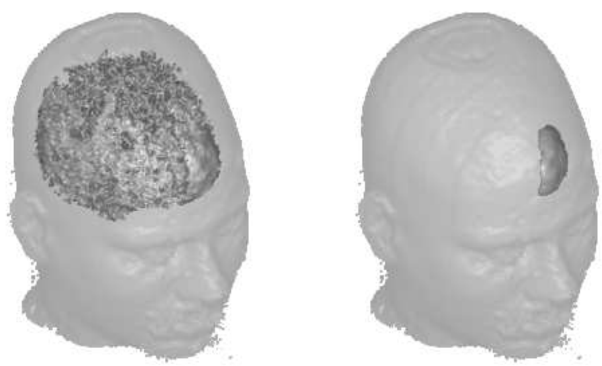


Figura 5.2: Ground truth and segmentation results for the 5 patients. The first row shows the ground truth provided by an specialist. The second row shows the intermediary classification result. The third row shows the final classification result after the outlier removal step. Patients 1 and 3 are children, and patient 1 has a visible anatomical deformity (top left).

We used the *leave-one-out* scheme [36] to classify the voxels of each patient (each patient's voxels were classified using the training sets from the other four patients). We used a radius of 5 voxels and a density threshold of 35% for the outlier removal step. With these parameters, we detected all 5 lesions, with no false positives. The voxel coverage of the lesions ranged from 62.5% to 90.3% (average: 76.9%). Fig. 5.2 shows some sample slices of ground truth, intermediary classification (after step (v)) and final classification. Even though the intermediary classification appears clean on 2D slices such as those shown in Fig. 5.2, a 3D renderization shows that it contains too much noise. Fig. 5.3 shows 3D renderizations of the intermediary (a) and final (b) lesion segmentations for patient 2.



(a) (b)

Figura 5.3: 3D renderizations of the lesion segmentation for patient 2: (a) intermediary classification and (b) segmentation after outlier removal.

The lesion segmentation method took about 30 minutes per patient on an Athlon64 3200+ PC: 1 minute for MSP location (which includes the brain segmentation), 20 seconds for CR computation, 28 minutes for feature extraction and RCE classification, and 40 seconds for the outlier removal step.

5.5 Conclusions

We presented a new method for segmentation of dysplastic lesions in MR-T1 images of the brain. Our method does not rely on template-based method and is applicable to any patient regardless of age or anatomical variations. We evaluated the method on 5 patients of ages 11y–51y, with good results on all of them. The average lesion coverage was 76.9%, providing a result slightly better than the state of the art [23].

Our method requires some adhoc parameters which were chosen experimentally, such as the radius and percentage values for outlier removal, and the parameters for texture features and RCE classifier training. We are currently working on automatic methods for choosing those parameters, as well as measuring their effect on the segmentation results. We are also working on obtaining more datasets in order to validate the method in a larger set of patients.

Capítulo 6

A Partitioned Algorithm for the Image Foresting Transform

6.1 Introduction

The Image Foresting Transform (IFT) [39] is a graph-based framework for the design and implementation of image processing operators. It reduces image processing operations, such as watersheds [96, 15], morphological reconstructions [42], skeletonization [41] and distance transforms [31], to the computation of a minimum-cost path forest over an implicit graph representation of the image. The IFT runs in linear time, but it does not take advantage of parallel and distributed computer systems. Its data structures also require considerable memory space [45], and this can be a limitation to the processing of large 3D images.

In this work we present the Partitioned IFT, an algorithm that computes any IFT as a set of independent IFTs over partitions of the input image. Both time and memory required to compute the IFT of each partition are proportional to the size of that partition. The minimum-cost path forests of the partitions are merged by fast differential IFTs [40]. This scheme provides the means to take advantage of parallel and distributed computer systems (by assigning each partition's IFT to a different central processing unit (CPU)) and to allow the computation of IFTs with a reduced memory footprint (by computing partition forests sequentially).

6.2 Related works

6.2.1 Related algorithms

Moga et al. [70] presented two parallel watershed algorithms that treat the image as a graph and perform independent flooding simulations in image partitions. Parallel flooding simulations are repeated while plateaus overflow to adjacent partitions. The same group [71] presented a similar parallel algorithm for the computation of the watershed-from-markers transform. Both works achieve scalable speedups in parallel architectures, but the speedup factor does not scale linearly with the number of processors. Moga et al. [70] achieve speedup factors[1] around 2 for 4-CPU systems, and 3.5 for 8-CPU systems. Bruno and Costa [18] present a distributed algorithm for the computation of Euclidean distance transforms (EDT) based on morphological dilations. Their algorithm achieves a speedup factor of 3.5 on a 4-CPU system.

6.2.2 The image foresting transform

The IFT algorithm is essentially Dijkstra's algorithm [1], modified for multiple sources and general path cost functions [39]. The image is interpreted as a directed graph whose nodes are the pixels. The edges are defined implicitly by an *adjacency relation* $\mathcal{A}$. Tree roots are drawn from a set $\mathcal{S}$ of *seed nodes* and path costs are given by a *path cost function* f. We use $P^*(s)$ to denote the current path reaching pixel s, $\langle s \rangle$ to denote a *trivial path* containing a single node, and $\langle s,t \rangle$ to denote the edge from pixel s to pixel t. $P^*(s) \cdot \langle s,t \rangle$ is the path that results from the concatenation of $P^*(s)$ and an edge $\langle s,t \rangle$.

The choice of $\mathcal{A}$, $\mathcal{S}$ and f define an IFT operator. The IFT algorithm can compute by ordered propagation any forest property that uses the seed set as reference. Usually, the IFT computes 4 maps: the cost map C stores the cost of the optimal path that reaches each pixel, the predecessor map P stores the predecessor of each pixel in the forest, the root map R stores the root of each pixel's optimal path, and the label map L stores object labels for each pixel. Algorithm 5 below computes the IFT.

Algorithm 5 – IFT

INPUT:	Image $\mathbf{I}$, Path-cost function f, Adjacency relation $\mathcal{A}$, Seed set $\mathcal{S}$ and Seed label map L_0.
OUTPUT:	Cost map C, Predecessor map P, Root map R and Label map L.
AUXILIARY:	Priority queue Q.

[1]The speedup factor of a parallel algorithm on an n-CPU parallel system is calculated as $\frac{t_1}{t_N}$, where t_1 is the time required to perform the computation on a single-CPU system, and t_N is the time required to perform the computation on an n-way system.

1. *Set* $Q \leftarrow \emptyset$.
2. **For each** *pixel* $s \in \mathbf{I} \setminus \mathcal{S}$, **do**
3.   *Set* $C(s) \leftarrow \infty$, $P(s) \leftarrow nil$, $R(s) \leftarrow s$ *and* $L(s) \leftarrow nil$.
4. **For each** *pixel* $s \in \mathcal{S}$, **do**
5.   *Set* $C(s) \leftarrow f(\langle s \rangle)$ *and* $L(s) \leftarrow L_0(s)$.
6.   *Insert* s *in* Q.
7. **While** $Q \neq \emptyset$, **do**
8.   *Remove a pixel* s *from* Q *such that* $C(s)$ *is minimum.*
9.   **For each** t *such that* $(s,t) \in \mathcal{A}$, **do**
10.   *Compute* $cost \leftarrow f(P^*(s) \cdot \langle s,t \rangle)$.
11.   **If** $cost < C(t)$ **then**
12.   **If** $t \in Q$ **then** *remove* t *from* Q.
13.   *Set* $P(t) \leftarrow s$, $C(t) \leftarrow cost$, $L(t) \leftarrow L(s)$, $R(t) \leftarrow R(s)$.
14.   *Insert* t *in* Q.

Lines 1–3 set the forest to an initial state where every node's optimum path is a trivial path with infinite cost. Lines 4–6 insert the seed pixels in the priority queue with a trivial path cost computed by f, and initialize seed labels for ordered propagation. The loop of lines 7–14 uses the priority queue to propagate the optimum paths and conquer the entire image. As long as f is finite and smooth [39], an optimum path with finite cost will be assigned to all pixels connected to $\mathcal{S}$. Once a pixel is removed from the queue (line 8), it is never inserted again. Therefore, the main loop is repeated $|\mathbf{I}|$ times. For integer path costs with limited increments, Q can be efficiently implemented such that insertions and removals take $O(1)$ time [1]. With small adjacency relations ($|\mathcal{A}| \ll |\mathbf{I}|$) and $O(1)$ queue operations, the IFT algorithm runs in $O(|\mathbf{I}|)$ time [39].

Two common path cost functions for IFT operators are f_{max} and f_{euc}, shown in Eqs. 6.1–6.2 below. Both f_{max} and f_{euc} are *smooth*, as required to ensure the correctness of the IFT [39].

$$f_{max}(\langle s_1, \ldots, s_n \rangle) = \begin{cases} max_{i=1}^{n} \left(I\left(s_i\right) \right) & \text{if } n > 1, \\ h(s_1) & \text{otherwise.} \end{cases} \tag{6.1}$$

$$f_{euc}(\langle s_1, \ldots, s_n \rangle) = \text{Euclidean distance between } s_1 \text{ and } s_n \tag{6.2}$$

where $I(s)$ is some value associated to pixel s (such as intensity or gradient intensity) and h is a handicap function for trivial paths. A watershed-from-markers transform can be implemented as an IFT where f is f_{max} (Eq. 6.1), $h = 0$ (for marker imposition), $\mathcal{A}$ is an adjacency with radius between 1 and $\sqrt{2}$ and $\mathcal{S}$ contains the watershed markers [39, 40]. A classical watershed can be implemented using $f = f_{max}$, $h(s) = I(s) + 1$ and $\mathcal{S} =$

I [65]. Function f_{euc} (Eq. 6.2) allows the computation of distance transforms [31], discrete Voronoi diagrams, skeletonizations and shape saliences [41, 93, 39, 10].

6.2.3 The differential image foresting transform

The differential IFT [40] (DIFT) was motivated by interactive 3D image segmentation applications where the user interactively selects the seed pixels. It is quite common for the user to add new seeds and remove previous ones based on the visualization of the segmentation result. The first IFT is computed by Algorithm 5 as usual, from a seed set $\mathcal{S}_0$. The maps C, P, R and L must be initialized to a forest of trivial paths with infinite costs before the first DIFT is computed. Given a set $\mathcal{S}'$ of seeds to be added and a set $\mathcal{S}''$ of tree roots to be removed, the DIFT computes the optimum path forest for the effective seed seet $\mathcal{S}_1 = (\mathcal{S}_0 \setminus \mathcal{S}'') \cup \mathcal{S}'$. The DIFT processes only pixels affected by the seed set editing, and runs in sublinear time. Instead of providing $\mathcal{S}''$ directly, the DIFT takes a set $\mathcal{M}$ of removal markers, and $\mathcal{S}''$ is computed as the set of roots of the pixels in $\mathcal{M}$. Algorithm 6 below is the main DIFT algorithm. The DIFT-TreeRemoval subroutine referenced in line 2 visits all pixels that belong to removed trees, sets their optimum paths to trivial paths with infinite costs (forcing their recalculation by Algorithm 6), and builds the set $\mathcal{F}$ of frontier pixels.

Algorithm 6 – DIFT

INPUT:	Image **I**, Cost map C, Predecessor map P, Root map R, Label map L, Path-cost function f, Adjacency relation $\mathcal{A}$, Set $\mathcal{S}'$ of new seed pixels, Set $\mathcal{M}$ of marking pixels, Seed label map L_0.
OUTPUT:	C, P, R and L.
AUXILIARY:	Priority queue Q, Frontier set $\mathcal{F}$.

1. *Set* $Q \leftarrow \emptyset$.
2. $(C, P, \mathcal{F}) \leftarrow$DIFT-TREEREMOVAL$(C, P, R, L, \mathcal{A}, \mathcal{M})$.
3. $\mathcal{F} \leftarrow \mathcal{F} \setminus \mathcal{S}'$.
4. **While** $\mathcal{S}' \neq \emptyset$, **do**
5.   *Remove any* t *from* $\mathcal{S}'$.
6.   **If** $f(\langle t \rangle) < C(t)$ **then**
7.   *Set* $C(t) \leftarrow f(\langle t \rangle)$, $R(t) \leftarrow t$, $L(t) \leftarrow L_0(t)$, $P(t) \leftarrow nil$.
8.   *Set* $\mathcal{F} \leftarrow \mathcal{F} \cup \{t\}$.
9. **While** $\mathcal{F} \neq \emptyset$, **do**
10.   *Remove any* t *from* $\mathcal{F}$ *and insert* t *in* Q.
11. **While** $Q \neq \emptyset$, **do**
12.   *Remove a pixel* s *from* Q, *such that* $C(s)$ *is minimum.*
13.   **For each** t *such that* $(s, t) \in \mathcal{A}$, **do**

14. *Compute cost* $\leftarrow f(P^*(s) \cdot \langle s, t \rangle)$.
15. **If** $cost < C(t)$ *or* $P(t) = s$ **then**
16. **If** $t \in Q$ **then** *remove* t *from* Q.
17. *Set* $P(t) \leftarrow s$, $C(t) \leftarrow cost$, $R(t) \leftarrow R(s)$, $L(t) \leftarrow L(s)$.
18. *Insert* t *in* Q.

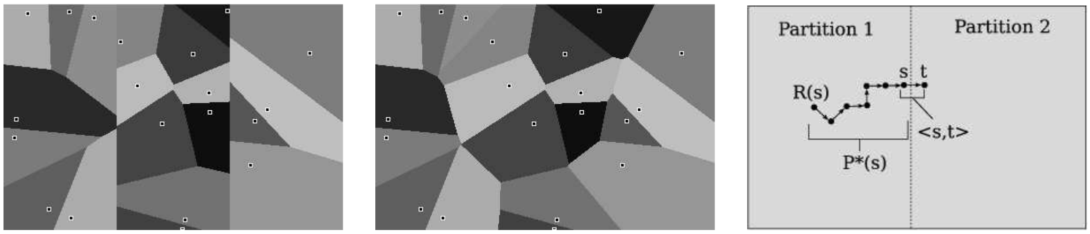
Lines 2–3 compute a set $\mathcal{F}$ of frontier pixels that belong to non-removed trees but share edges with pixels in removed trees. Lines 4–10 insert the new seeds and the frontier pixels in the queue. Lines 11–18 are very much like the main loop of the IFT Algorithm (Algorithm 5), except for the condition $P(t) = s$ in line 14, which forces the update of all pixels that had their optimum paths modified. The result of the DIFT is an optimum path forest for the "effective seed set" $\mathcal{S}_1 = (\mathcal{S}_0 \setminus \mathcal{S}'') \cup \mathcal{S}'$.

6.3 The partitioned image foresting transform

In the Partitioned IFT (PIFT), we split the input image and seed set in N_P *partitions*. The number of partitions can be chosen to match the number of available processing nodes, or so that the computer system has enough memory to run the IFT algorithm on each image partition. Partitions do not need to be equally sized. We compute independent IFTs on each partition. At this point, we have an optimum forest that ignores the inter-partition edges of the graph. Figure 6.1a shows an example of this partial result for the EDT using a set of random pixels as seeds and 3 partitions. To allow propagation through the inter-partition graph edges, we consider the paths obtained by the concatenation of each edge $\langle s, t \rangle$ to $P^*(s)$ (Figure 6.1c). When $f(P^*(s) \cdot \langle s, t \rangle)$ is less than the current cost of t, or the edge was part of the destination pixel's previous optimal path, the endpoint is added as seed in a differential IFT so that it can be propagated. If more than one inter-partition edge share a same endpoint t, the one that provides the lower path cost $P^*(t)$ is propagated. A new iteration of differential IFTs is computed for each partition. The PIFT halts when no inter-partition edge satisfies the criteria for addition. Figure 6.1b shows the complete EDT, obtained after 2 iterations over the 3 partitions.

The differential IFTs used in the Partitioned IFT always have an empty set of removal markers. The Partition-IFT algorithm below (Algorithm 7) computes the IFT within a partition. It is essentially the differential IFT algorithm without tree removal, and with special treatment of inter-partition edges.

Algorithm 7 – Partition-IFT



(a) (b) (c)

Figura 6.1: Labels of an EDT with the Partitioned IFT: (a) Partial result after the first iteration and (b) final result after the second iteration. (c) PIFT notation: $\langle s,t\rangle$ is an inter-partition edge, $P^*(s)$ is the optimum path assigned to s, and $R(s)$ the root of $P^*(s)$.

Input:	Image partition $\mathbf{I}'$, Cost map C, Predecessor map P, Root map R, Label map L, Path-cost function f, Adjacency relation $\mathcal{A}$, Set $\mathcal{S}$ of seed pixels, Seed label map L_0, Set $\mathcal{E}_I$ of incoming inter-partition edges.
Output:	Maps C, P, R, L and Set $\mathcal{E}_O$ of outgoing inter-partition edges.
Auxiliary:	Priority queue Q.

Set $Q \leftarrow \emptyset$, $\mathcal{E}_O \leftarrow \emptyset$.
If $\mathcal{S} \neq \emptyset$ **then**
    For each *pixel* $s \in \mathbf{I}'$, **do**
        Set $C(s) \leftarrow \infty$, $P(s) \leftarrow nil$, $R(s) \leftarrow s$ *and* $L(s) \leftarrow nil$.
    For each *pixel* $s \in \mathcal{S}$, **do**
        Set $C(s) \leftarrow f(\langle s\rangle)$ *and* $L(s) \leftarrow L_0(s)$.
        Insert s *in* Q.
For each *edge* $\langle s,t\rangle \in \mathcal{E}_I$, **do**
    Compute cost $\leftarrow f(P^*(s) \cdot \langle s,t\rangle)$.
    If *cost* $< C(t)$ *or* $P(t) = s$ **then**
        Set $C(t) \leftarrow cost$, $P(t) \leftarrow s$, $R(t) \leftarrow R(s)$ *and* $L(t) \leftarrow L(s)$.
        Insert t *in* Q.
While $Q \neq \emptyset$, **do**
    Remove a pixel s *from* Q, *such that* $C(s)$ *is minimum.*
    For each t *such that* $(s,t) \in \mathcal{A}$, **do**
        If $t \in \mathbf{I}'$ **then**
            Compute cost $\leftarrow f(P^*(s) \cdot \langle s,t\rangle)$.
            If *cost* $< C(t)$ *or* $P(t) = s$ **then**
                If $t \in Q$ **then** *remove* t *from* Q.
                Set $P(t) \leftarrow s$, $C(t) \leftarrow cost$, $R(t) \leftarrow R(s)$, $L(t) \leftarrow L(s)$.
                Insert t *in* Q.
        Else *Insert* $\langle s,t\rangle$ *in* $\mathcal{E}_O$.

The DIFT is unable to tell whether the algorithm is on the first iteration, therefore the initial state of the forest must be set before the first iteration. In the PIFT, the seed set $\mathcal{S}$ will only be non-empty in the first iteration. We use this property to initialize the partition's forest to trivial paths with infinite costs in lines 2–4. Lines 5–7 queue and initialize the seed pixels in the same way the IFT does. Lines 8–12 process the incoming inter-partition edges $\mathcal{E}_I$. Edges that offer lower costs to their endpoints or belonged to the previous forest are queued for propagation. If multiple edges in $\mathcal{E}_I$ reach the same endpoint, the edge that provides the lower cost for the endpoint takes precedence. The main loop in lines 13–22 is very similar to the main loop of the DIFT, with the addition of the partition test $t \in \mathbf{I}'$ in line 16. Edges within the current partition are processed normally. Inter-partition edges are added to the outgoing edge set $\mathcal{E}_O$ (line 22). Note that the cost computation in line 9 may require additional information about $P^*(s)$, which can contain pixels of several partitions. All path information required to compute $f(P^*(s) \cdot \langle s, t \rangle)$ must be passed along with the set $\mathcal{E}_I$. For f_{max}, only $C(s)$ is required. For f_{euc}, only $R(s)$ is required. Since $L(s)$ may be propagated in line 11, it must also be part of the input. Passing each element of $\mathcal{E}_I$ as $\{s, t, C(s), R(s), L(s)\}$ is enough to compute the PIFT with either f_{max} or f_{euc}. The PIFT algorithm (Algorithm 8) that computes the IFT of an image $\mathbf{I}$ from its partitions is shown below.

Algorithm 8 – Partitioned IFT

INPUT:	Image $\mathbf{I}$, Path-cost function f, Adjacency relation $\mathcal{A}$, Set $\mathcal{S}$ of seed pixels, Seed label map L_0, Number of partitions N_P.
OUTPUT:	Cost map C, Predecessor map P, Root map R, Label map L.
AUXILIARY:	Edge sets $\mathcal{E}$, $\mathcal{E}'$, $\mathcal{E}''$ and $\mathcal{E}'''$, Seed set $\mathcal{S}'$.

1. *Set* $\mathcal{E} \leftarrow \emptyset$.
2. *Split* $\mathbf{I}$ *in* N_P *partitions* $\mathbf{I}[1] \ldots \mathbf{I}[N_P]$.
3. **For** $i = 1$ **to** N_P, **do**
4.   *Set* $\mathcal{S}' = \{s \mid s \in \mathcal{S} \wedge s \in \mathbf{I}[i]\}$.
5.   *Set* $(C[i], P[i], R[i], L[i], \mathcal{E}') \leftarrow$ PARTITION-IFT$(\mathbf{I}[i], C[i], P[i], R[i], L[i], f, \mathcal{A}, \mathcal{S}', L_0, \emptyset)$.
6.   *Set* $\mathcal{E} \leftarrow \mathcal{E} \cup \mathcal{E}'$.
7. **Repeat**
8.   *Set* $\mathcal{E}''' \leftarrow \emptyset$.
9.   **For** $i = 1$ **to** N_P, **do**
10.   *Set* $\mathcal{E}'' = \{\langle s, t \rangle \mid \langle s, t \rangle \in \mathcal{E} \wedge t \in \mathbf{I}[i]\}$.
11.   *Set* $(C[i], P[i], R[i], L[i], \mathcal{E}') \leftarrow$ PARTITION-IFT$(\mathbf{I}[i], C[i], P[i], R[i], L[i], f, \mathcal{A}, \emptyset, nil, \mathcal{E}'')$.

12. $\quad\quad$ *Set* $\mathcal{E}''' \leftarrow \mathcal{E}''' \cup \mathcal{E}'$.
13. $\quad$ *Set* $\mathcal{E} \leftarrow \mathcal{E}'''$.
14. $\quad$ **Until** $\mathcal{E} = \emptyset$.
15. *Set* $C \leftarrow \cup_{i=1}^{N_P} C[i]$, $P \leftarrow \cup_{i=1}^{N_P} P[i]$, $R \leftarrow \cup_{i=1}^{N_P} R[i]$ *and* $L \leftarrow \cup_{i=1}^{N_P} L[i]$.

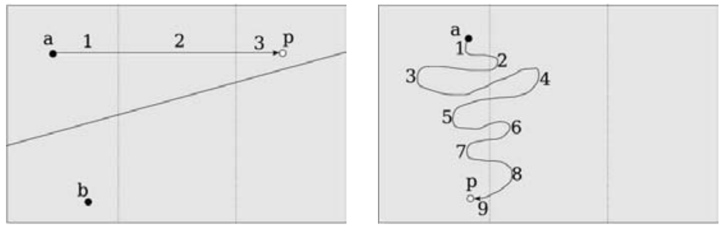
Lines 1–2 initialize the inter-partition edge set $\mathcal{E}$ and split the input image in N_P partitions. The loop in lines 3–6 run the first IFT iteration on each partition. All inter-partition edges are accumulated in the set $\mathcal{E}$. The loop in lines 7–14 run the remaining IFT iterations on the partitions, until no propagation occurs and the set $\mathcal{E}$ of inter-partition edges is empty (line 14).

For parallel architectures, both loops (lines 3–6 and 7–14) can be done in parallel. For distributed systems, the executions of Partition-IFT (Algorithm 7) can be performed as remote procedure calls. Note that the partitioned maps ($C[i]$, $P[i]$, $R[i]$ and $L[i]$) are only needed at the end of the algorithm, to compose the final IFT maps. In a distributed implementation, these maps can be kept on the remote processing nodes and do not need to be transferred at each call to Partition-IFT, as they are not modified by the caller.

Performance Considerations. The overall number of pixels processed by the PIFT is larger than $|\mathbf{I}|$. After the loop of lines 3–6 of Algorithm 8, the PIFT has already processed $|\mathbf{I}|$ nodes. However, the number of pixels processed by the loop of lines 7–14 decreases at each iteration, and the algorithm converges rapidly to the optimum path forest. The number of PIFT iterations — i.e., one iteration of the loop of lines 3–6 plus the number of iterations of the loop of lines 7–14 — is bounded by the maximum number of inter-partition edges contained by an optimum path, plus one. Each inter-partition edge postpones the resolution of the optimum path to the next PIFT iteration. Figure 6.2 illustrates some examples. In an Euclidean distance transform (Figure 6.2a), all paths flow away from the roots, and a path may cross at most $N_P - 1$ partition boundaries, requiring at most N_P PIFT iterations. For path cost functions like f_{max}, there is no restriction to the shape of optimum paths, and cases like the one in Figure 6.2b can occur. However, as the number of iterations increases, the number of pixels processed by each iteration tends to decrease, and the PIFT converges more rapidly to the optimum forest.

6.4 Experimental results

We implemented the PIFT as a client-server system, with a simple TCP stream-based protocol for communication between the master client that executes Algorithm 8 and the



(a) (b)

Figura 6.2: Partition crossings and PIFT iterations: In the PIFT-EDT, paths cross at most $N_P - 1$ partition boundaries. In (a), $P^*(p)$ crosses 2 boundaries to reach p from a. The numbers are the iteration in which the path segment is propagated. (b) For general path-cost functions, a path may cross partition boundaries several times.

distributed servers that execute Algorithm 7. In our implementation, the image is always split in equal-sized partitions, using the x coordinate to separate partitions (such as in Figure 6.1a). We chose 3 applications to evaluate the PIFT:

1. WS-BRAIN: Watershed-based segmentation of a 3D MR image of the brain, using $f = f_{max}$ and $\mathcal{A}$ =6-neighborhood adjacency. Seeds were selected interactively in the background and in the brain. The gradient intensity was computed by a Gaussian enhacement filter followed by morphological gradient computation [40]. The size of the image is $356 \times 356 \times 241$, with a voxel size of $0.70mm^3$ (Figure 6.3a–c).

2. EDT-RND: Euclidean distance transform of 1000 random points within a 256^3 volume, using $f = f_{euc}$ and $\mathcal{A}$ =26-neighborhood (Figure 6.3d).

3. EDT-BRAIN: Euclidean distance transform using the border of the brain object (segmented in the first application) as seed set ($|\mathcal{S}| = 355,556$). $f = f_{euc}$, $\mathcal{A}$ =26-neighborhood and volume size is $356 \times 356 \times 241$ (Figure 6.3e).

First, we measured the processing overhead of the PIFT as the number of partitions (N_P) increases. We computed the 3 applications with the PIFT, using from 1 to 10 partitions. Table 6.1 and Figure 6.4 present the number of nodes processed in each case and the upper bound for the speedup factor. These results indicate that a 10-way parallel system may be able to offer a speedup factor of 6.60 to the EDT computation, and a factor of 2.34 to the Watershed transform on these instances of problems.

The EDT computations required at most 4 iterations before halting. PIFTs based on f_{max} are less efficient, since they allow free-form paths that can traverse several partitions. This can be noticed by the irregularity and increased slope of the plot in Figure 6.4b, as compared to Figure 6.4a. The WS-BRAIN PIFTs required at most 23 iterations to

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

Figura 6.3: Images from the evaluation applications: (a) Slice from the WS-BRAIN input image. (b) gradient intensity of (a). (c) 3D renderization of the WS-BRAIN result. (d) Visualization of the discrete Voronoi diagram, result of the EDT-RND. (e) Slice from the distance map computed in EDT-BRAIN.

converge. The number of processed nodes grows linearly with the number of partitions. In real data with non-uniform distributions (WS-BRAIN and EDT-BRAIN), bad choices of partition boundaries may increase the number of processed nodes, such as in the $N_P = 4$ and $N_P = 8$ cases of EDT-BRAIN and $N_P = 5$ of WS-BRAIN.

Tabela 6.1: Number of processed nodes and upper bound for the speedup factor in each application, using up to 10 partitions.

N_P	**WS-BRAIN**		**EDT-RND**		**EDT-BRAIN**	
	Nodes	Speedup	Nodes	Speedup	Nodes	Speedup
1	30.5×10^6	1.00	16.8×10^6	1.00	30.5×10^6	1.00
2	48.6×10^6	1.25	17.3×10^6	1.94	31.5×10^6	1.93
3	59.0×10^6	1.55	17.7×10^6	2.84	34.2×10^6	2.67
4	62.7×10^6	1.94	18.2×10^6	3.69	39.6×10^6	3.08
5	76.7×10^6	1.98	18.6×10^6	4.51	37.8×10^6	4.03
6	75.1×10^6	2.43	19.2×10^6	5.25	38.7×10^6	4.72
7	92.2×10^6	2.31	19.7×10^6	5.96	42.8×10^6	4.98
8	98.1×10^6	2.48	20.1×10^6	6.68	46.8×10^6	5.21
9	106.5×10^6	2.57	20.5×10^6	7.37	42.2×10^6	6.50
10	130.2×10^6	2.34	21.0×10^6	8.00	46.2×10^6	6.60

In a second set of experiments we used the PIFT to compute EDT-RND, EDT-BRAIN and WS-BRAIN in two parallel systems: a PC with 2 CPUs (Athlon MP 1800+@1150 MHz) and 2 GB of RAM, and a Compaq AlphaServer GS140 6/525 with 10 CPUs (Alpha EV6@525 MHz) and 8 GB of RAM. Table 6.2 presents the results. On the EDT applications, we achieved speedup factors very close to the measured upper bounds (Table 6.1) for $N_P = 2$ and $N_P = 4$. On other hand, there was little or no speedup for

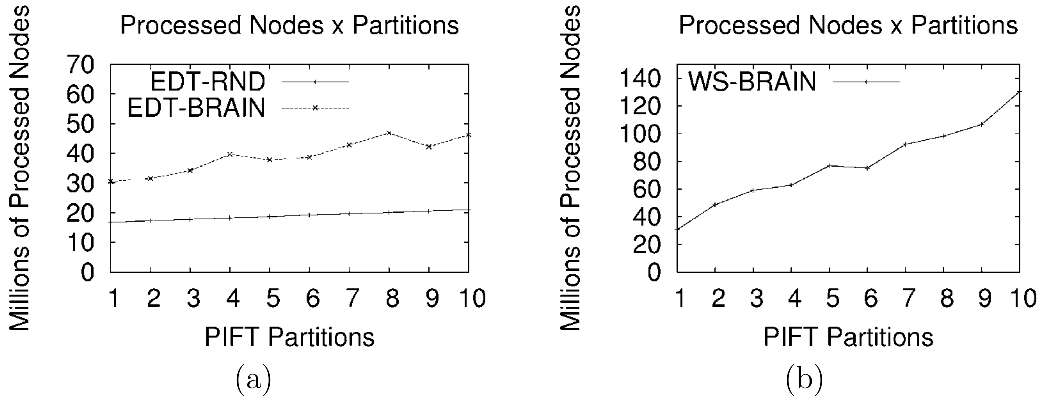


Figura 6.4: Number of processed nodes vs. number of partitions for (a) EDT-RND, EDT-BRAIN and (b) WS-BRAIN.

Tabela 6.2: PIFT performance on two parallel computer systems. Times are given in seconds.

System	N_P	**WS-BRAIN**		**EDT-RND**		**EDT-BRAIN**	
		Time	Speedup	Time	Speedup	Time	Speedup
Dual Athlon	1	258.1	1.00	195.9	1.00	459.5	1.00
	2	242.9	1.06	106.2	1.84	246.3	1.87
10-CPU GS140	1	280.6	1.00	228.8	1.00	611.4	1.00
	2	284.3	0.99	126.6	1.81	324.2	1.89
	4	226.2	1.24	73.0	3.13	274.3	2.23
	8	249.3	1.13	49.3	4.64	214.1	2.86
	10	336.4	0.83	47.9	4.78	197.7	3.09

the watershed application. Our prototype implementation uses a naive communication protocol with no data compression. Besides that, the edge set transfers of lines 5 and 11 of Algorithm 8 were implemented in a sequential way, and instances with a large number of partitions and/or a large number of PIFT iterations (such as WS-BRAIN with $N_P = 10$) performed poorly because the CPUs remained idle while waiting for the client to complete the sequential edge set transfers.

6.5 Conclusion and future works

We introduced the Partitioned Image Foresting Transform, an algorithm that computes minimum-cost path forests as a set of independent DIFTs [39, 40] in partitions of the input image. The PIFT is useful for taking advantage of parallel computer systems and for computing IFTs in computer systems with limited memory, such as handhelds and embedded systems. The PIFT is applicable to any IFT-based operator, and therefore can be readily employed to parallelize morphological reconstructions [42], watershed transforms [96, 15, 40, 10], distance transforms [31, 39] and skeletonizations [41, 93], among other operators. It is a trend in microprocessor technology to compensate CPU speed limitations by producing multi-core CPUs. The PIFT is an important contribution that allows existing image processing applications to use modern hardware efficiently with minimum effort.

We implemented a prototype PIFT system with a simple client-server architecture built on top of TCP streams. Even with no data compression and with some inneficient network operations, we achieved speedup factors very close to the expected upper bounds for EDT operations. PIFT-based watershed segmentation performed poorly due to the inneficiency of edge set transfers in our prototype. With a better protocol, the PIFT should be able to reach speedup factors closer to the upper bounds in Table 6.1.

Future works include: development of better protocols for implementation of the PIFT in parallel systems, evaluation of the speedup bounds for specific operators – such as the watershed transform – and investigation of enhancements to the PIFT such as partitioning schemes and iteration scheduling among nodes.

Capítulo 7

Conclusões e Trabalhos Futuros

7.1 Contribuições

Esta tese desenvolveu um método para segmentação de displasias corticais focais em imagens volumétricas de ressonância magnética T1. Displasias corticais focais são as malformações mais comuns em casos de epilepsia refratária, e seu diagnóstico e localização são passos cruciais para o planejamento do tratamento. Este método torna automático um procedimento que era tedioso e subjetivo [72], realizando uma análise totalmente automatizada das imagens de RM e apontando para o especialista as regiões onde há indícios de lesões de FCD.

O método proposto obteve detecção de 100% das lesões, com cobertura de 76,9% dos voxels lesionais. O método foi avaliado em um grupo de 5 pacientes (todos com FCDs no lobo frontal), por restrição de tempo e recursos computacionais. Em condições normais de uso, com um grupo fechado de treinamento, o método requer 30 minutos em um PC de 3.2 GHz. Nas condições de avaliação *leave-one-out* [36], em que um treinamento diferente precisa ser montado para cada paciente (de forma a excluir treinamentos realizados no próprio paciente testado), o método exigiu em torno de 4 horas de processamento por paciente. A avaliação *leave-one-out* foi escolhida por ter sido usada também na avaliação do principal trabalho nesta área [23].

O treinamento do classificador RCE leva tempo proporcional ao quadrado do número de amostras de textura. Embora tivéssemos 14 delineações de FCDs frontais, o tempo de processamento para realizar *leave-one-out* em todas as 14 com RCE seria proibitivo com os recursos computacionais disponíveis no período da pesquisa. Durante a pesquisa realizamos tentativas de classificação com classificadores k-NN e com redes neurais artificiais [36], com pouco sucesso. A disparidade numérica entre amostras lesionais e saudáveis levou a resultados ruins com k-NN, e redes neurais artificiais mostraram-se muito instáveis em relação aos *outliers* comuns nesta aplicação.

Experimentos com lesões não frontais mostraram que suas características de textura diferem das lesões de FCD frontais. Como não obtivemos uma quantidade significativa de pacientes com displasias ocipitais, temporais e parietais, não foi possível avaliar adequadamente o método em displasias não frontais.

Avaliações ainda não publicadas do nosso método indicam que o aumento do conjunto de treinamento não afeta negativamente a taxa de cobertura. Uma avaliação do método em cérebros saudáveis (controles) indica que ele não detecta componentes conexos de falsos positivos com volumes significativos.

Para chegar ao método apresentado, esta tese apresentou novas soluções para segmentação automatizada de imagens [12], localização do plano de simetria do cérebro [14], cálculo automático da reformatação curvilinear [10] e apresentou uma solução de aprendizado supervisionado baseada em análise de assimetria de textura [13].

Esta tese também apresentou uma proposta de paralelização/particionamento da IFT [39, 11]. Embora ainda haja muitas possibilidades de aperfeiçoamento nesta técnica, este trabalho demonstra que o projeto de operadores baseados na IFT oferece flexibilidade para contornar limitações de hardware (capacidade de processamento e capacidade de memória).

Como resultado desta pesquisa, realizamos a demarcação de FCDs em cérebros de 20 pacientes, cada um deles com uma única lesão. Destas 20 lesões, 14 estão localizadas no lobo frontal, 2 no lobo ocipital, 3 no lobo parietal e 1 no lobo temporal. Os pacientes delineados estão detalhados no Apêndice B. A aparência das lesões varia de acordo com a localização, e não houve um número adequado de pacientes para compor grupos de treinamento para os lobos ocipital, parietal e temporal (onde FCDs são naturalmente mais raras).

7.2 Outras Contribuições

Enquanto pesquisávamos métodos de segmentação, desenvolvemos e publicamos o conceito de competição de sementes κ-conexas [38]. Este trabalho, entretanto, mostrou-se menos robusto para a segmentação de imagens médicas tri-dimensionais do que a poda de árvores [12], levando-nos a optar pela poda de árvores para o método de segmentação de displasias. A técnica de competição κ-conexa continua sendo ativamente aperfeiçoada pelos demais contribuidores deste trabalho, para outras aplicações de segmentação de imagens.

Durante toda a nossa pesquisa com aplicações interativas de análise de imagens médicas, utilizamos técnicas de projeção de voxels para realizar as tarefas de visualização [8]. Sempre tivemos a preocupação de poder aproveitar a capacidade dos novas soluções de aceleração gráfica, que utilizam representações poligonais. Realizamos duas publicações [34, 33] onde avaliamos a eficiência de usar malhas poligonais deformáveis para visualizar os objetos (cérebro, envelope do cérebro lesões segmentadas). A manutenção de malhas poligonais requer uma representação dual que consome mais memória e torna a programação das aplicações um pouco mais complexa, além de exigir *drivers* proprietários nem sempre prontamente disponíveis ou legalmente desejáveis em sistemas operacionais de código livre. Até o momento da conclusão desta tese, consideramos que a representação por malhas deformáveis não supera a simplicidade e praticidade da projeção por voxels, e continuamos utilizando o método de projeção de voxels descrito no Capítulo 6 de [8].

7.3 Extensões Futuras

Há diversas oportunidades para aperfeiçoar a pesquisa na área desegmentação de FCDs, dentre as quais:

- Avaliar outras técnicas de aprendizado supervisionado, tais como o Optimum Path Forest [76] desenvolvido recentemente.
- Realizar segmentação de substância branca e cinzenta (WM/GM), avaliar como as displasias afetam a qualidade desta segmentação, e utilizá-la para tomar outras medidas ignoradas pelo método atual, tal como a espessura do córtex.
- Avaliar o método em um número maior de pacientes e controles.
- Delinear lesões temporais, ocipitais e parietais, para formar uma base de dados mais abrangente.
- Implementar ferramentas com interface amigável para treino, classificação e visualização dos resultados.

Em relação à IFT particionada [11], também há oportunidades claras para desenvolvimento de trabalhos de pesquisa:

- Avaliar outros operadores além de Watershed e transformada de distância Euclideana.
- Propor novas estratégias de particionamento, de acordo com a distribuição das sementes e/ou propriedades da imagem.
- Realizar implementações mais eficientes, com melhor utilização dos canais de comunicação entre processadores – redes locais, memória compartilhada, etc.

Apêndice A

Ferramenta de Software: BrainTag

Neste apêndice apresentamos o principal software implementado para realizar o pré-processamento e delineamento das lesões.

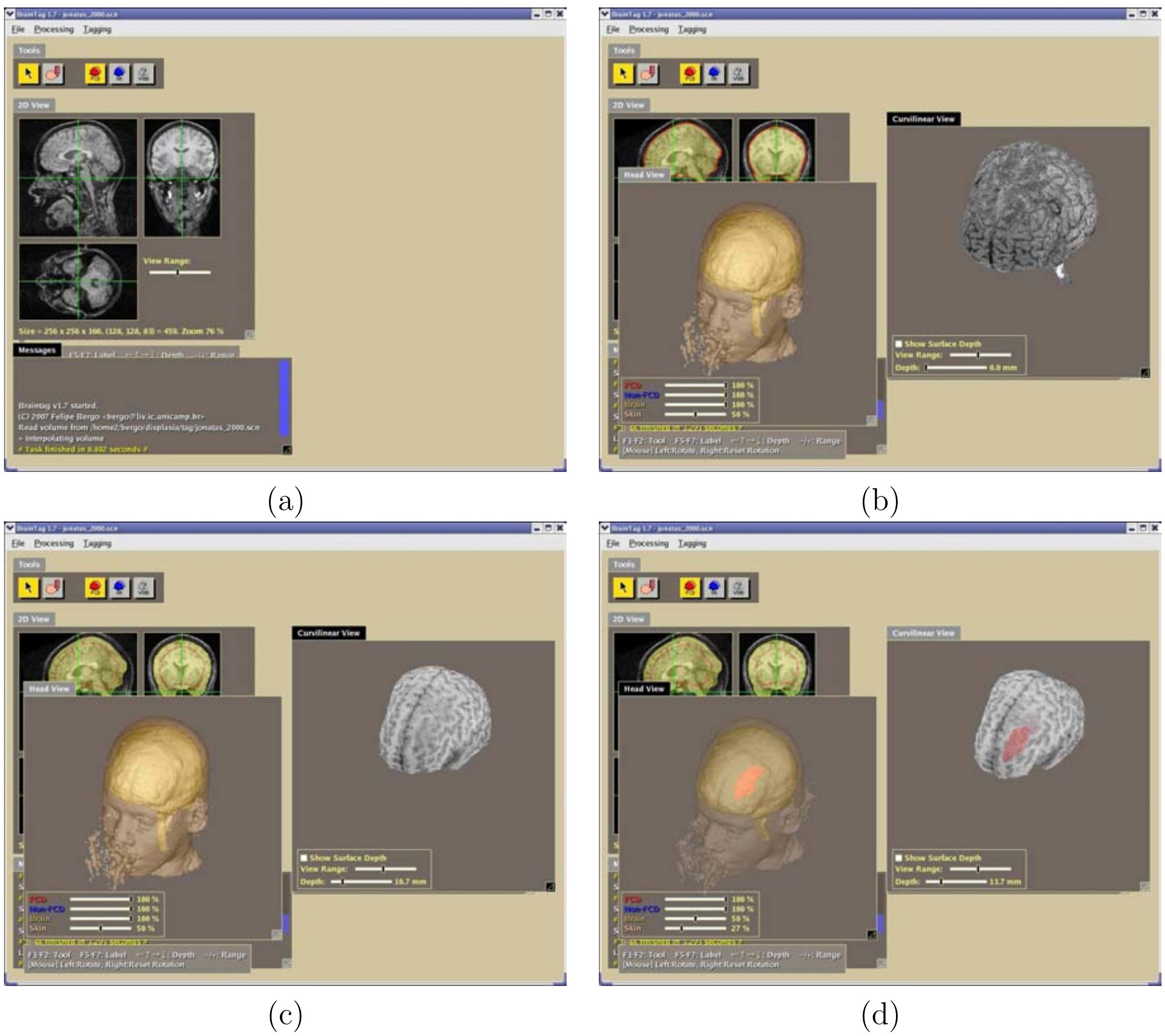
A conversão entre formatos DICOM (formato de saída da RM) e SCN (formato usado para processamento) é realizada pelo software IVS [9], desenvolvido no meu trabalho de mestrado [40, 8]. Após a segmentação das displasias, também utilizamos o IVS para visualizar os resultados da segmentação superpostos à imagem original de RM.

A.1 Braintag

O software *Braintag* incorpora os algoritmos de segmentação automática do cérebro por poda de árvores [12], localização do plano inter-hemisférico [14], cálculo automático da reformatação curvilinear [10] e permite que o especialista delineie interativamente as lesões sobre a visualização da reformatação curvilinear.

A Figura A.1(a) mostra a tela do software após carregar um volume de ressonância magnética. Um clique em uma opção de menu (Processing, Prepare Volume for Tagging) dispara todo o processamento necessário – segmentação do cérebro, localização do plano inter-hemisférico e reformatação curvilinear, sem a necessidade de qualquer parâmetro. As Figuras A.1(b–c) mostram telas do software ao terminar o processamento, permitindo que o operador navegue pelas reformatações curvilineares e delineie as lesões interativamente (Figura A.1(d)).

O BrainTag foi implementado em linguagem C++ em ambiente Linux, utilizando o toolkit gráfico GTK+ (versão 2 da API). Seu código, incluindo todos os algoritmos implementados, tem em torno de 10000 linhas de código. BrainTag lê e salva os dados volumétricos no formato SCN, utilizado por todas as ferramentas de análise de imagens desenvolvidas pelo grupo de pesquisa do Prof. Alexandre Falcão. A renderização 3D usa o método de projeção de voxels descrito no capítulo 6 de [8]. A preparação do volume para marcação requer de 1 a 2 minutos de processamento para um volume de RM típico, em um PC de 3 GHz.



(a) (b) (c) (d)

Figura A.1: (a) BrainTag após carregar um volume de RM. (b–c) BrainTag após computar a reformatação curvilinear. (d) BrainTag mostrando o delineamento de uma lesão e sua renderização tri-dimensional.

Apêndice B

Pacientes Delineados

Neste apêndice apresentamos as informações de sexo, idade e localização das lesões de FCD na base de 20 pacientes com lesões delineadas neste trabalho. As lesões foram delineadas pela Dra. Clarissa Yasuda, sob supervisão do Prof. Dr. Fernando Cendes.

Paciente	**Idade/Sexo**	**Localização**	**Data do Exame**
#1	11/M	Frontal Esquerdo	2002.04.22
#2	34/M	Ocipital Direito	1999.03.24
#3	16/M	Parietal Esquerdo	2002.08.21
#4	29/M	Parietal Direito	1998.07.29
#5	29/F	Ocipital Direito	1999.11.23
#6	24/F	Frontal Direito	2000.11.28
#7	12/M	Frontal Esquerdo	2000.06.28
#8	51/M	Frontal Esquerdo	2004.01.26
#9	30/F	Frontal Direito	2003.10.22
#10	34/M	Frontal Esquerdo	1998.08.04
#11	8/M	Frontal Direito	2002.07.23
#12	18/F	Frontal Direito	1999.09.21
#13	7/M	Parietal Esquerdo	2000.05.03
#14	17/M	Frontal Esquedo	1999.04.07
#15	9/F	Frontal Direito	2000.06.06
#16	25/M	Frontal Direito	1998.06.23
#17	24/M	Temporal Esquerdo	2006.03.29
#18	32/F	Frontal Direito	1999.10.06
#19	42/F	Frontal Direito	1998.07.15
#20	24/F	Frontal Direito	2006.01.08

Tabela B.1: Pacientes com FCDs delineadas.

Referências Bibliográficas

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice-Hall, 1993.

[2] S. B. Antel, D. L. Collins, N. Bernasconi, F. Andermann, R. Shinghal, R. E. Kearney, D. L. Arnold, and A. Bernasconi. Automated detection of focal cortical dysplasia lesions using computational models of their MRI characteristics and texture analysis. *NeuroImage*, 19(4):1748–1759, Aug 2003.

[3] B. Ardekani, J. Kershaw, M. Braun, and Iwao Kanno. Automatic detection of the mid-sagittal plane in 3-D brain images. *IEEE Trans. on Medical Imaging*, 16(6):947–952, Dec 1997.

[4] R. Audigier, R. A. Lotufo, and A. X. Falcão. 3D visualization to assist iterative object definition from medical images. *Computerized Medical Imaging and Graphics*, 30(4):217–230, 2006.

[5] A.J. Barkovich, H.A. Rowley, and F. Andermann. MR in partial epilepsy: value of high-resolution volumetric techniques. *American Journal of Neuroradiology*, 16:339–343, February 1995.

[6] T. R. Barrick, C. E. Mackay, S. Prima, F. Maes, D. Vandermeulen, T. J. Crow, and N. Roberts. Automatic analysis of cerebral asymmetry: na exploratory study of the relationship between brain torque and planum temporale asymmetry. *NeuroImage*, 24(3):678–691, Feb 2005.

[7] A. C. Bastos, R. M. Comeau, F. Andermann, D. Melanson, F. Cendes, F. Dubeau, S. Fontaine, D. Tampieri, and A. Olivier. Diagnosis of subtle focal dysplastic lesions: Curvilinear reformatting from three-dimensional magnetic resonance imaging. *Annals of Neurology*, 46(1):88–94, 1999.

[8] F. P. G. Bergo. Segmentação interativa de volumes baseada em regiões. Master's thesis, Universidade Estadual de Campinas, Instituto de Computação, Feb 2004.

[9] F. P. G. Bergo and A. X. Falcão. IVS (Interactive Volume Segmentation). http://www.liv.ic.unicamp.br/%7Ebergo/ivs, 2004.

[10] F. P. G. Bergo and A. X. Falcão. Fast and automatic curvilinear reformatting of MR images of the brain for diagnosis of dysplastic lesions. In *Proc. of 3rd Intl. Symp. on Biomedical Imaging*, pages 486–489. IEEE, Apr 2006.

[11] F. P. G. Bergo and A. X. Falcão. A partitioned algorithm for the image foresting transform. In *Proc. 8th Intl. Symposium Mathematical Morphology (ISMM'07)*, pages 425–436, Rio de Janeiro, Brazil, Oct 2007. INPE.

[12] F. P. G. Bergo, A. X. Falcão, P. A. V. Miranda, and L. M. Rocha. Automatic image segmentation by tree pruning. *J Math Imaging and Vision*, 29(2–3):141–162, Nov 2007.

[13] F. P. G. Bergo, A. X. Falcão, C. L. Yasuda, and F. Cendes. FCD segmentation using texture asymmetry of MR-T1 images of the brain. In *Proc. of 5th Intl. Symp. on Biomedical Imaging*, Paris, France, May 2008. IEEE.

[14] F. P. G. Bergo, G. C. S. Ruppert, L. F. Pinto, and A. X. Falcão. Fast and robust mid-sagittal plane location in 3D MR images of the brain. In *Proc. BIOSIGNALS 2008 – Intl. Conf. on Bio-Inspired Syst. and Sig. Proc.*, pages 92–99, Funchal, Portugal, Jan 2008. Springer/IEEE.

[15] S. Beucher and F. Meyer. The morphological approach to segmentation: The watershed transformation. In *Mathematical Morphology in Image Processing*, chapter 12, pages 433–481. Marcel Dekker, 1993.

[16] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 105–112, 2001.

[17] M. E. Brummer. Hough transform detection of the longitudinal fissure in tomographic head images. *IEEE Trans. on Medical Imaging*, 10(1):66–73, Mar 1991.

[18] O. M. Bruno and L. F. Costa. A parallel implementation of exact euclidean distance transform based on exact dilations. *Microprocessors and Microsystems*, 28(3):107–113, Apr 2004.

[19] G. Bueno, O. Musse, F. Heitz, and J. P. Armspach. Three-dimensional segmentation of anatomical structures in MR images on large data bases. *Magnetic Resonance Imaging*, 19:73–88, 2001.

[20] L. D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 53(2):211–218, 1991.

[21] D. L. Collins, A. P. Zijdenbos, V. Kollokian, J. G. Sled, N. J. Kabani, C. J. Holmes, and A. C. Evans. Design and construction of a realistic digital brain phantom. *IEEE Trans. on Medical Imaging*, 17(3):463–468, Jun 1998.

[22] O. Colliot, N. Bernasconi, N. Khalili, S. B. Antel, V. Naessens, and A. Bernasconi. Individual voxel-based analysis of gray matter in focal cortical dysplasia. *NeuroImage*, 29(1):162–171, Jan 2006.

[23] O. Colliot, T. Mansi, N. Bernasconi, V. Naessens, D. Klironomos, and A. Bernasconi. Segmentation of focal cortical dysplasia lesions on MRI using level set evolution. *NeuroImage*, 32(4):1621–1630, Oct 2006.

[24] N. Colombo, L. Tassi, C. Galli, A. Citterio, G. Lo Russo, G. Scialfa, and R. Spreafico. Focal cortical dysplasias: MR imaging, histopathologic, and clinical correlations in surgically treated patients with epilepsy. *American Journal of Neuroradiology*, 24:724–733, April 2003.

[25] T. Cootes, G. Edwards, and C.J. Taylor. Active appearance models. In *European Conference on Computer Vision (ECCV)*, volume 2, pages 484–498, 1998.

[26] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models – their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[27] T. J. Crow. Schizophrenia as an anomaly of cerebral asymmetry. In K. Maurer, editor, *Imaging of the Brain in Psychiatry and Related Fields*, pages 3–17. Springer, 1993.

[28] J. G. Csernansky, S. Joshi, L. Wang, J. W. Haller, M. Gado, J. P. Miller, U. Grenander, and M. I. Miller. Hippocampal morphometry in schizophrenia by high dimensional brain mapping. *Proceedings of the National Academy os Sciences of the United States of America*, 95(19):11406–11411, Sep 1998.

[29] J. G. Csernansky, L. Wang, S. Joshi, J. P. Miller, M. Gado, D. Kido, D. McKeel, J. C. Morris, and M. I. Miller. Early DAT is distinguished from aging by high-dimensional mapping of the hippocampus. *Neurology*, 55:1636–1643, Dec 2000.

[30] M. B. Cuadra, C. Pollo, A. Bardera, O. Cuisenaire, J-G. Villemure, and J-P. Thiran. Atlas-based segmentation of pathological MR brain images using a model of lesion growth. *IEEE Trans Medical Imaging*, 23(10):1301–1314, Oct 2004.

[31] P.E. Danielsson. Euclidean distance mapping. *Computer Graphics and Image Processing*, 14:227–248, 1980.

[32] R. J. Davidson and K. Hugdahl. *Brain Asymmetry*. MIT Press/Bradford Books, 1996.

[33] F. de Goes, , F. P. G. Bergo, A. X. Falcão, S. Goldenstein, and L. Velho. Adapted dynamic meshes for deformable surfaces. In *XIX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 213–220, Manaus, Brazil, Oct 2006. IEEE.

[34] F. de Goes, , F. P. G. Bergo, A. X. Falcão, S. Goldenstein, and L. Velho. Dynamic meshes for deformable surfaces. In *SIGGRAPH 2006*, page 88, Boston, MA, Aug 2006. ACM.

[35] E.R. Dougherty and R.A. Lotufo. *Hands-on Morphological Image Processing*. SPIE Press, Bellingham, WA, 2003.

[36] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

[37] A. X. Falcão, F. P. G. Bergo, and P. A. V. Miranda. Image segmentation by tree pruning. In *XVII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 65–71. IEEE, Oct 2004.

[38] A. X. Falcão, P. A. V. Miranda, A. Rocha, and F. P. G. Bergo. Object detection by k-connected seed competition. In *XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 97–104. IEEE, Oct 2005.

[39] A. X. Falcão, J. Stolfi, and R. A. Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(1):19–29, 2004.

[40] A.X. Falcão and F.P.G. Bergo. Interactive volume segmentation with differential image foresting transforms. *IEEE Trans. on Medical Imaging*, 23(9):1100–1108, 2004.

[41] A.X. Falcão, L.F. Costa, and B.S. da Cunha. Multiscale skeletons by image foresting transform and its applications to neuromorphometry. *Pattern Recognition*, 35(7):1571–1582, 2002.

[42] A.X. Falcão, B.S. da Cunha, and R.A. Lotufo. Design of connected operators using the image foresting transform. In *Proc. of SPIE on Medical Imaging*, volume 4322, pages 468–479, Feb 2001.

[43] A.X. Falcão, J.K. Udupa, and F.K. Miyazawa. An ultra-fast user-steered image segmentation paradigm: Live-wire-on-the-fly. *IEEE Trans. on Medical Imaging*, 19(1):55–62, 2000.

[44] A.X. Falcão, J.K. Udupa, S. Samarasekera, S. Sharma, B.E. Hirsch, and R.A. Lotufo. User-steered image segmentation paradigms: Live-wire and live-lane. *Graphical Models and Image Processing*, 60(4):233–260, Jul 1998.

[45] P. Felkel, M. Bruckschwaiger, and R. Wegenkittl. Implementation and complexity of the watershed-from-markers algorithm computed as a minimal cost forest. *Computer Graphics Forum (EUROGRAPHICS)*, 20(3):(C) 26–35, 2001.

[46] L. Ford and D. Fulkerson. *Flows in networks.* Princeton University Press, 1962.

[47] R.S.J. Frackowiak, K.J. Friston, C. Frith, R. Dolan, C.J. Price, S. Zeki, J. Ashburner, and W.D. Penny. *Human Brain Function.* Academic Press, 2nd edition, 2003.

[48] J. L. Frater, R. A. Prayson, H. H. Morris III, and W. E. Bingaman. Surgical pathologic findings of extratemporal-based intractable epilepsy. *Arch Pathol Lab Med*, 124(4):545–549, Apr 2000.

[49] N. Geschwind and W. Levitsky. Human brain: Left-right asymmetries in temporale speech region. *Science*, 161(3837):186–187, Jul 1968.

[50] V. Grau, A.U.J. Mewes, M. Alcaniz, R. Kikinis, and S.K. Warfield. Improved watershed transform for medical image segmentation using prior information. *IEEE Trans. on Medical Imaging*, 23(4):447–458, Apr 2004.

[51] R. Guillemaud, P. Marais, A. Zisserman, B. McDonald, T. J. Crow, and M. Brady. A three dimensional mid sagittal plane for brain asymmetry measurement. *Schizophrenia Research*, 18(2–3):183–184, Feb 1996.

[52] R. Haralick, K. Shanmugam, and I. Dinstein. Texture parameters for image classification. *IEEE Trans. Syst. Man Cyb. 3*, pages 610–621, 1973.

[53] S. Haykin. *Neural Networks: A comprehensive foundation.* Prentice-Hall, 1998.

[54] J. R. Highley, L. E. DeLisi, N. Roberts, J. A. Webb, M. Relja, K. Razi, and T. J. Crow. Sex-dependent effects of schizophrenia: an MRI study of gyral folding, and cortical and white matter volume. *Psychiatry Research: Neuroimaging*, 124(1):11–23, Sep 2003.

[55] R. E. Hogan, K. E. Mark, I. Choudhuri, L. Wang, S. Joshi, M. I. Miller, and R. D. Bucholz. Magnetic resonance imaging deformation-based segmentation of the hippocampus in patients with mesial temporal sclerosis and temporal lobe epilepsy. *J. Digital Imaging*, 13(2):217–218, May 2000.

[56] Q. Hu and W. L. Nowinski. A rapid algorithm for robust and automatic extraction of the midsagittal plane of the human cerebrum from neuroimages based on local symmetry and outlier removal. *NeuroImage*, 20(4):2153–2165, Dec 2003.

[57] L. Junck, J. G. Moen, G. D. Hutchins, M. B. Brown, and D. E. Kuhl. Correlation methods for the centering, rotation, and alignment of functional brain images. *The Journal of Nuclear Medicine*, 31(7):1220–1226, Jul 1990.

[58] I. Kapouleas, A. Alavi, W. M. Alves, R. E. Gur, and D. W. Weiss. Registration of three dimensional MR and PET images of the human brain without markers. *Radiology*, 181(3):731–739, Dec 1991.

[59] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.

[60] J. Kassubek, H. J. Huppertz, J. Spreer, and A. Schulze-Bonhage. Detection and localication of focal cortical dysplasia by voxel-based 3-D MRI analysis. *Epilepsia*, 43(6):596–602, Jun 2002.

[61] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(2):147–159, Feb 2004.

[62] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[63] Y. Liu, R. T. Collins, and W. E. Rothfus. Robust midsagittal plane extraction from normal and pathological 3D neuroradiology images. *IEEE Trans. on Medical Imaging*, 20(3):175–192, Mar 2001.

[64] Y. Liu, L. A. Teverovskiy, O. L. Lopez, H. Aizenstein, C. C. Meltzer, and J. T. Becker. Discovery of biomarkers for alzheimer's disease prediction from structural

MR images. In *2007 IEEE Intl. Symp. on Biomedical Imaging*, pages 1344–1347. IEEE, April 2007.

[65] 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*, volume 18, pages 341–350. Kluwer, Jun 2000.

[66] R.A. Lotufo, A.X. Falcão, and F. Zampirolli. IFT-Watershed from gray-scale marker. In *Proc. of XV Brazilian Symp. on Computer Graphics and Image Processing*, pages 146–152. IEEE, Oct 2002.

[67] C. E. Mackay, T. R. Barrick, N. Roberts, L. E. DeLisi, F. Maes, D. Vandermeulen, and T. J. Crow. Application of a new image analysis technique to study brain asymmetry in schizophrenia. *Psychiatry Research*, 124(1):25–35, Sep 2003.

[68] S. Minoshima, K. L. Berger, K. S. Lee, and M. A. Mintun. An automated method for rotational correction and centering of three-dimensional functional brain images. *The Journal of Nuclear Medicine*, 33(8):1579–1585, 1992.

[69] P. A. V. Miranda, F. P. G. Bergo, L. M. Rocha, and A. X. Falcão. Tree-pruning: A new algorithm and its comparative analysis with the watershed transform for automatic image segmentation. In *Proc. of the XIX Brazillian Symposium on Computer Graphics and Image Processing*, pages 37–44. IEEE, Oct 2006.

[70] A. N. Moga, B. Cramariuc, and M. Gabbouj. Parallel watershed transformation algorithms for image segmentation. *Parallel Computing*, 24(14):1981–2001, Dec 1998.

[71] A. N. Moga and M. Gabbouj. Parallel marker-based image segmentation with watershed transformation. *Journal of Parallel and Distributed Computing*, 51(1):27–45, May 1998.

[72] M. A. Montenegro, L. M. Li, M. M. Guerreiro, C. A. M. Guerreiro, and F. Cendes. Focal cortical dysplasia: Improving diagnosis and localization with magnetic resonance imaging multiplanar and curvilinear reconstruction. *J Neuroimg*, 12(3):224–230, Jul 2002.

[73] L.G. Nyul, A.X. Falcão, and J.K. Udupa. Fuzzy-connected 3D image segmentation at interactive speeds. *Graphical Models*, 64(5):259–281, 2003.

[74] N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9:62–66, Mar 1979.

[75] A. Palmini, F. Andermann, A. Olivier, D. Tampieri, Y. Robitaille, E. Andermann, and G. Wright. Focal neuronal migration disorders and intractable partial epilepsy: A study of 30 patients. *Annals of Neurology*, 30(6):741–749, Dec 1991.

[76] J. P. Papa, A. X. Falcão, P. A. V. Miranda, and C. T. N. Suzuki. Design of robust pattern classifiers nased on optimum-path forests. In *Proc. 8th Intl. Symposium Mathematical Morphology (ISMM'07)*, pages 337–348, Rio de Janeiro, Brazil, Oct 2007. INPE.

[77] S. Prima, S. Ourselin, and N. Ayache. Computation of the mid-sagittal plane in 3D brain images. *IEEE Trans. on Medical Imaging*, 21(2):122–138, Feb 2002.

[78] L. Bischof R. Adams. Seeded region growing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994.

[79] Rogue Research. BrainSight. http://www.rogue-research.com/epilepsy.htm.

[80] J.B.T.M. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41:187–228, 2000.

[81] P. M. Ruggieri, I. Najm, R. Bronen, M. Campos, F. Cendes, J. S. Duncan, H. G. Weiser, and W. H. Theodore. Neuroimaging of the cortical dysplasias. *Neurology*, 62:S27–S29, 2004.

[82] 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*, 82:42–56, 2001.

[83] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug 2000.

[84] S. M. Sisodiya. Surgery for malformations of cortical development causing epilepsy. *Brain*, 123(6):1075–1091, Jun 2000.

[85] S. M. Smith and M. Jenkinson. Accurate robust symmetry estimation. In *Proc MICCAI '99*, pages 308–317, London, UK, 1999. Springer-Verlag.

[86] S. Srivastava, F. Maes, D. Vandermeulen, W. van Paesschen, P. Dupont, and P. Suetens. Feature-based statistical analysis of structural MR data for automatic detection of focal cortical dysplastic lesions. *NeuroImage*, 27(2):253–266, Aug 2005.

[87] M. Styner and G. Gerig. Hybrid boundary-medial shape description for biologically variable shapes. In *Proc. of IEEE Workshop on Mathematical Methods in Biomedical Imaging Analysis (MMBIA)*, pages 235–242. IEEE, 2000.

[88] C. Sun and J. Sherrah. 3D symmetry detection using the extended Gaussian image. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(2):164–168, Feb 1997.

[89] J. Talairach and P. Tournoux. *Co-Planar Steriotaxic Atlas of the Human Brain.* Thieme Medical Publishers, 1988.

[90] D. C. Taylor, M. A. Falconer, C. J. Bruton, and J. A. Corsellis. Focal dysplasia of the cerebral cortex in epilepsy. *J Neurol Neurosurg Psychiatry*, 34(4):369–387, Aug 1971.

[91] L. Teverovskiy and Y. Liu. Truly 3D midsagittal plane extraction for robust neuroimage registration. In *Proc. of 3rd IEEE Intl. Symp. on Biomedical Imaging*, pages 860–863. IEEE, April 2006.

[92] R. S. Torres and A. X. Falcão. Contour salience descriptors for effective image retrieval and analysis. *Image and Vision Computing*, 25(1):3–13, Jan 2007.

[93] R. S. Torres, A. X. Falcão, and L. F. Costa. A graph-based approach for multiscale shape analysis. *Pattern Recognition*, 37(6):1163–1174, 2004.

[94] A. V. Tuzikov, O. Colliot, and I. Bloch. Evaluation of the symmetry plane in 3D MR brain images. *Pattern Recognition Letters*, 24(14):2219–2233, Oct 2003.

[95] J.K. Udupa and S. Samarasekera. Fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation. *Graphical Models and Image Processing*, 58:246–261, 1996.

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

[97] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–511–I–518, 2001.

[98] I. Volkau, K. N. B. Prakash, A. Ananthasubramaniam, A. Aziz, and W. L. Nowinski. Extraction of the midsagittal plane from morphological neuroimages using the Kullback-Leibler's measure. *Medical Image Analysis*, 10(6):863–874, Dec 2006.

[99] L. Wang, S. C. Joshi, M. I. Miller, and J. G. Csernansky. Statistical analysis of hippocampal asymmetry in schizophrenia. *NeuroImage*, 14(3):531–545, Sep 2001.

[100] W-C. Wu, C-C. Huang, H-W. Chung, M. Liou, C-J. Hsueh, C-S. Lee, M-L. Wu, , and C-Y. Chen. Hippocampal alterations in children with temporal lobe epilepsy with or without a history of febrile convulsions: Evaluations with MR volumetry and proton MR spectroscopy. *AJNR Am J Neuroradiol*, 26(5):1270–1275, May 2005.

[101] D. Zheng, Y. Zhao, and J. Wang. An efficient method of license plate location. *Pattern Recognition Letters*, 26:2431–2438, Nov 2005.