

Um Marcador de Imagens do Glioblastoma Multiforme (GBM) para apoio na Simulação de Tratamento

Vinicius B. Zuchi¹, Lucas P. Pinheiro¹, Aurélio G. Neto¹,
Augusto R. Amaral², Gilmário B. Santos³, Claudio C. Sá³

¹Aluno do Departamento de Ciência da Computação (DCC)
Universidade do Estado de Santa Catarina (UDESC)
Joinville - SC - Brasil

²Aluno do Dep. de Medicina - Univ. da Região de Joinville (UNIVILLE)
Joinville - SC - Brasil

³Professor do Departamento de Ciência da Computação (DCC)
Universidade do Estado de Santa Catarina (UDESC)
Joinville - SC - Brasil

lucas_p@outlook.com

{liquimaster, grott.aurelio, augusto.radunz}@gmail.com

{claudio.sa, gilmario.santos}@udesc.br

Abstract. *The Glioblastoma Multiforme (GBM) is a type of malignant and very aggressive brain tumor. For this reason, computational tools to support the GBM study are of great interest to the field of biomedical research. This paper describes the construction of an image marker software based on image processing for to assist the user in the graphical segmentation of the region of the tumor in the image. The development of the program presented here was supervised by medical specialist and its use was essential to start the simulation of the evolution of the GBM undergoing treatment with the drug bevacizumab.*

1. Introdução

Seja nos consultórios, seja nos laboratórios de pesquisa, as aplicações computacionais na área médica têm se tornado comuns no cotidiano da prática da medicina. Dentre as várias possibilidades de aplicações computacionais na área médica encontram-se as simulações. Os sistemas de simulação proveem mecanismos para avaliação da progressão do tratamento de uma doença baseando-se em dados estatísticos e procedimentos computacionais realizados a partir de modelos matemáticos. No seu nível mais avançado, esse tipo de sistema permite a obtenção de informações de prognóstico da evolução da doença no que é especialmente importante para os casos de doenças agressivas. Uma dessas doenças é o GBM ou glioblastoma multiforme, um tumor do sistema nervoso central muito agressivo (nível IV) e de ocorrência mais comum em adultos [Guedes 2010; Grossman et al. 2010; Holland 2000].

O presente trabalho está inserido no escopo do projeto de um simulador que utiliza autômatos celulares para a representação computacional da evolução do GBM sob tratamento com a droga bevacizumabe (*Avastin*[®]). A descrição detalhada desse simulador encontra-se no trabalho de Pinheiro (2015). A medicação com bevacizumabe visa

enfraquecer ou eliminar o tumor pela inibição da criação de novos vasos sanguíneos, os quais nutririam as células tumorais [Monteagudo e Santos 2014; Friedman et al. 2009]. Dessa forma, esse artigo apresenta um software editor de imagens de GBM com a funcionalidade necessária à inicialização do simulador descrito em Pinheiro (2015).

Esse artigo está estruturado conforme descrito a seguir: na seção 2 é descrito o contexto da aplicação da ferramenta demarcadora, na seção 3 explica-se os fundamentos de processamento de imagens aplicados. Por sua vez, a seção 4 é dedicada à descrição das funções da interface do programa, as bibliotecas e ambientes de programação utilizadas são descritas na seção 5, enquanto que na seção 6 são apresentados os resultados do uso do programa proposto e feitas análises pertinentes. Por fim, na seção 7 é feita a conclusão desse trabalho.

2. Contexto de Aplicação do Editor

Na atual fase do desenvolvimento do sistema de simulação foram feitos testes para 14 casos do GBM associados a pacientes diferentes. Cada caso se constituiu de exames de imagem, selecionados do respectivo volume tomográfico, sem a identificação do paciente, obtido com o apoio de profissional médico.

Para os testes do simulador, considerando um caso qualquer, para a inicialização do simulador é necessário obter a região do GBM na imagem pré-tratamento. Por outro lado, para aferir os resultados do simulador é necessário obter dados sobre a região do GBM pós-tratamento. Isso pode ser observado na Figura 1.

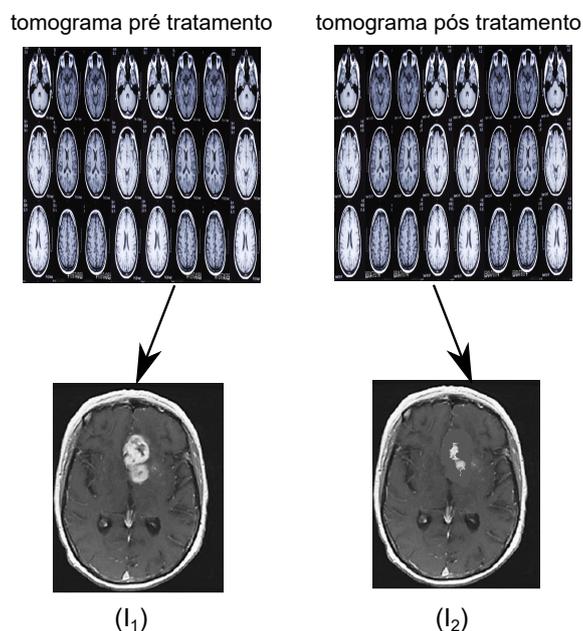


Figura 1. Para avaliar a simulação de um caso se faz necessário selecionar um par de imagens, pré-tratamento e pós-tratamento, a partir dos respectivos volumes tomográficos (tomogramas).

Foi desenvolvido um editor gráfico, um marcador para a seleção de regiões de interesse nas imagens (Figura 4-A), esse editor permite a criação de um arquivo representando uma região específica da imagem, no caso, regiões que demarcam o estágio do

GBM. Na metodologia empregada para testes do sistema de simulação, para cada caso foi selecionado um par de imagens (I_1, I_2) as quais capturaram dois estágios da doença (GBM):

- (I_1) no estágio anterior ao início tratamento e
- (I_2) no estágio posterior ao ciclo de tratamento com bevacizumabe por um determinado número de dias.

Considerando um caso, um operador/especialista demarca a região do GBM em cada par de imagens (I_1, I_2) criando um conjunto de dados. Nesse conjunto se inclui o par de arquivos *png* (I_1^i, I_2^o) cujos conteúdos correspondem às respectivas regiões do GBM correspondentes às fases pré-tratamento e pós-tratamento. O primeiro arquivo I_1^i (fase pré-tratamento) serve como base para especificar a inicialização da simulação, enquanto o segundo I_2^o (fase pós-tratamento) serve como função objetivo, para ser utilizado no processo de avaliação do resultado da simulação, para o caso em análise. Esse esquema pode ser visto na Figura 2.

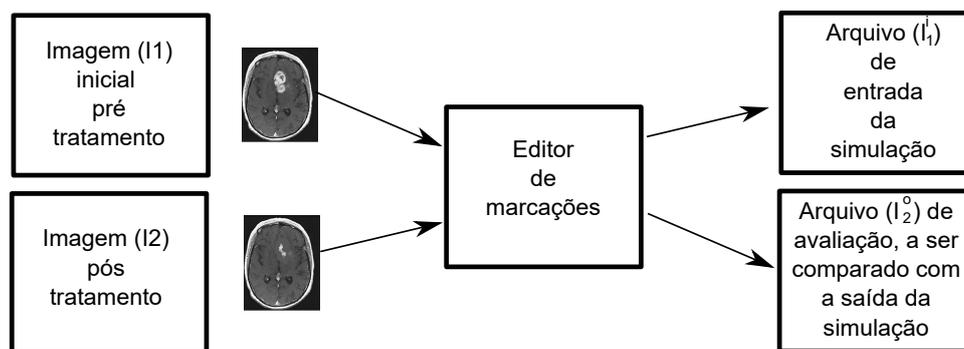


Figura 2. Contexto de uso do editor de imagens de GBM.

2.1. Modelo Matemático da Simulação

ACs são modelos matemáticos compostos por um conjunto de células, onde o estado de cada célula é atualizado de acordo com regras locais e relações de vizinhança entre as mesmas [Wolfram 1984]. De forma geral, ACs são constituídos por um conjunto de estados, um conjunto de células, um tipo de vizinhança e uma regra de transição. O conjunto de estados é representado por Σ e tem tamanho k . Cada célula é associada a um índice i , e o estado da célula i no tempo t é representado por S_i^t , com $S_i^t \in \Sigma$. A vizinhança η_i^t é composta pelo estado da célula i e pelos estados de suas vizinhas. A regra de transição $\phi(\eta_i^t)$ determina qual o estado resultante para uma célula de acordo com a sua vizinhança η_i^t [Oliveira 2003]

As células são dispostas de acordo com a quantidade de dimensões usadas no modelo. Por exemplo: em um AC unidimensional (1D), todas as células ficam dispostas em uma linha. Já em um AC bidimensional (2D), as células ficam dispostas em um plano. Cada célula deve assumir um (e apenas um) estado em cada instante do tempo. A cada instante do tempo, as células atualizam seus estados de acordo com a regra de transição, sendo que esta atualização é síncrona, ou seja, todas as células atualizam seus estados ao mesmo tempo.

ACs podem ser combinados com outros modelos computacionais, resultando em modelos híbridos capazes de simular comportamentos mais complexos. Um desses modelos são as equações de reação-difusão. Essas equações descrevem as mudanças nas concentrações de uma ou mais substâncias distribuídas no espaço com o passar do tempo. Esse modelo leva em conta dois processos: a reação, a qual faz com que as substâncias se transformem umas nas outras e a difusão, a qual faz com que as substâncias se espalhem pelo espaço, indo de áreas com maior concentração para áreas com menor concentração.

A modelagem deste comportamento pode ser feita por uma única equação diferencial no seguinte formato [Ruuth 1995]:

$$\frac{\partial w}{\partial t} = F(w) + D\Delta w, \quad (1)$$

onde $w = w(x, t)$ é o vetor de concentração das substâncias na posição x no tempo t , F é uma função que representa as reações, D é a matriz diagonal com os coeficientes de difusão das substâncias e Δ é o operador de Laplace. A variável x representa um vetor posição, ou seja, o espaço considerado pode possuir qualquer número de dimensões. Nesta equação, o primeiro termo ($F(w)$) modela as reações, enquanto o segundo ($D\Delta w$) modela a difusão.

O modelo aqui adotado é um híbrido de autômato celular com equações de reação-difusão. O AC é usado para simular os estados das células tumorais e vasos sanguíneos, enquanto as equações de reação-difusão são utilizadas para simular a distribuição de oxigênio, glicose e bevacizumabe.

O AC é bidimensional e utiliza a vizinhança de Neumann [Kari 2005], onde cada célula possui como vizinhas as células à esquerda, à direita, abaixo e acima. Como o AC é bidimensional, as suas células são dispostas dentro de uma área retangular. O conjunto de estados é composto pelos estados de célula proliferativa, célula quiescente, célula necrótica, vaso sanguíneo e espaço vazio.

3. Fundamentação Teórica

Algumas funções do editor de imagens utilizam técnicas e/ou conceitos de processamento de imagens, as quais são abordadas nessa seção:

3.1. Imagem Binária

Uma imagem pode ser compreendida como uma função $f(x, y)$. Para as imagens em tons de cinza (monocromáticas), essa função a qual associa cada ponto do espaço-imagem 2D a uma intensidade de cinza. Para as imagens coloridas o usual é utilizar-se de três canais de intensidades de cores primárias (RGB) para representar a cor de um pixel ($R(x,y), G(x,y), B(x,y)$). A imagem binária, por sua vez, corresponde a uma função que associa um pixel a apenas dois valores: preto ou branco (zero ou um).

3.2. Conexidade

O conceito de conexidade (ou conectividade) é importante para a determinação de regiões de pixels vizinhos e similares, portanto, o conceito de vizinhança e a determinação de um critério de similaridade são as bases para a determinação da conexidade de pixels e regiões em uma imagem [Pedrini e Schwartz 2008]. Os requisitos de similaridade podem ser de

diferentes naturezas, tais como intensidade de cinza, cor ou textura. Quanto à vizinhança, existem duas categorias: a vizinhança V4 e a V8. Tomando-se uma matriz quadrada $M_{3,3}$ de nove células e considerando W como seu elemento central, o conjunto de vizinhos V4 de W corresponderiam a todos os elementos localizados, acima, abaixo, à direita e à esquerda de W . Quanto ao conjunto de vizinhos V8, esse corresponderia a união do conjunto V4 e todas as outras posições de $M_{3,3}$ que cercam W .

3.3. Ruído em Imagem

Uma outra interpretação de uma imagem a equivale a um sinal 2D discretizado. Imagens, portanto, são sujeitas à presença de elementos espúrios chamados ruídos, os quais são gerados por diversos fatores, sendo os mais importantes aqueles causados por problemas na aquisição da imagem, perturbações climáticas, ambientais, presença de luz inadequada (apresentando *flickering* por exemplo) ou problemas gerados durante a transmissão da imagem. A presença de ruído normalmente é indesejada porque perturba a informação útil na imagem.

3.3.1. Filtro Passa-Baixa Gaussiano

Sendo sujeitas a ruídos, as imagens podem ser filtradas por processos semelhantes àqueles utilizados em telecomunicações, para a filtragem de sinais transmitidos visando otimizar a sua qualidade na recepção. Dessa forma, imagens contêm frequências e quanto mais variável for a informação de cor/intensidade em uma imagem, mais altas serão as frequências que ela conterá. Por outro lado, quanto mais regiões homogêneas ela apresentar, mais baixas frequências ela conterá. Ruídos aleatórios são elementos de alta frequência, um operador bastante utilizado para amortecê-los é o filtro gaussiano (passa-baixas), onde a imagem filtrada será $G(x, y)$ exibida na Eq. 2.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (2)$$

3.4. Limiarização

Segundo Pedrini e Schwartz (2008) a limiarização consiste na classificação dos pixels de uma imagem de acordo com a especificação de limiar(es). Para segmentar objetos (regiões de pixels) na imagem pode ser definido um limiar T_1 a ser utilizado para selecionar pixels na imagem original, podendo inclusive produzir uma imagem binária de saída (Equação 3). Outros limiares também podem ser introduzidos, de maneira a selecionar fatias mais específicas de intensidades (ou cores) de pixels, é o caso do limiar secundário T_2 na Equação 4. Nesse caso, todos os pixels $h(x, y)$ na imagem de saída tais que $T_1 < f(x, y) \leq T_2$ receberão valor igual a 1.

$$h(x, y) = \begin{cases} 0, & \text{se } f(x, y) \leq T_1, \\ 1, & \text{se } f(x, y) > T_1 \end{cases} \quad (3)$$

$$h(x, y) = \begin{cases} 0, & \text{se } f(x, y) \leq T_1 \text{ ou } f(x, y) > T_2, \\ 1, & \text{caso contrário} \end{cases} \quad (4)$$

4. Descrição da Ferramenta de Demarcação

O demarcador de imagens funciona basicamente como uma interface de funções e tem como objetivo servir de auxílio para análise de tomografias. Essa interface é capaz de salvar as demarcações feitas na imagem original, e além disso, informações adicionais a respeito da região demarcada também são salvas, para um processamento posterior.

Por ter sido projetada para ser o mais minimalista possível e por servir a um propósito consideravelmente específico, a interface é constituída somente de ferramentas e funções que foram julgadas necessárias e suficientes para o funcionamento adequado do programa.

Todas as funções são utilizadas através de dispositivo mouse. A funcionalidade do editor de imagens é destacada na Figura 4, cada função da interface está descrita abaixo:

- A) Marcação Livre: seleção da área de interesse (GBM) por meio de um lápis virtual. Enquanto o mouse estiver sendo movido com o botão esquerdo pressionado, uma curva demarca a região de interesse na imagem, continuamente. A região será delimitada por esta curva. O desenho da curva se encerra quando o botão esquerdo for liberado.
- B) Demarcação por Linhas: seleção da área de interesse (GBM) por meio da junção de segmentos de retas criadas entre dois cliques de mouse. Ao clicar em uma certa região da imagem, a ferramenta desenhará uma linha entre o último e o penúltimo ponto clicado, e manterá a posição destes pontos, caso alguma das demarcações necessite ser desfeita/movida.
- C) Demarcação por Retângulo: seleção de área por meio de um retângulo. Ao clicar e arrastar o mouse, uma borda retangular vermelha será desenhada demarcando uma região.
- D) Preenchimento por Cor: A partir de um pixel inicial selecionado pelo usuário através do mouse, por meio do critério de conectividade, a ferramenta pinta todos os vizinhos conexos desse pixel até preencher uma região delimitada por uma borda vermelha.
- E) Apaga uma região preenchendo-a com uma cor selecionada.
- F) Paleta para a escolha das cores (preto, branco e vermelho) que são processadas pelo programa.
- G) Processar Região Demarcada: Esta função permitirá ao usuário visualizar com uma maior clareza a região demarcada. Primeiramente, toda a região que está fora da área retangular de demarcação é marcada com a cor preta. Terminada a primeira etapa, a função verifica os valores dos pixels dentro da área retangular e aplica uma limiarização (*thresholding*) com um valor escolhido pelo usuário, com uma única diferença de que os pixels vermelhos são mantidos. Caso o usuário deixe marcado a opção de limiar secundário, é aplicado um segundo limiar nos pixels vizinhos aos que passaram no primeiro limiar, com a diferença de que o segundo é aplicado com um valor diferente. Ao final da execução da função, haverá somente pixels vermelhos, pretos e brancos na imagem.
- H) Detecção Automática do Crânio: Se for importante para uma melhor localização das estruturas envolvidas, pode ser útil lançar mão dessa ferramenta. A detecção do crânio é feita quando o usuário abre uma nova imagem, esta operação é dividida em quatro partes:

- i) é aplicado um filtro gaussiano sobre a imagem original, a fim de reduzir o ruído;
- ii) sobre a imagem filtrada, é aplicado um segundo filtro para detectar as bordas;
- iii) são procurados pixels candidatos à fazerem parte do crânio e, quando achados, o restante dos pixels pertencentes ao crânio são adicionados por conectividade;
- iv) os pixels que foram salvos como pertencentes ao crânio são demarcados na imagem original.

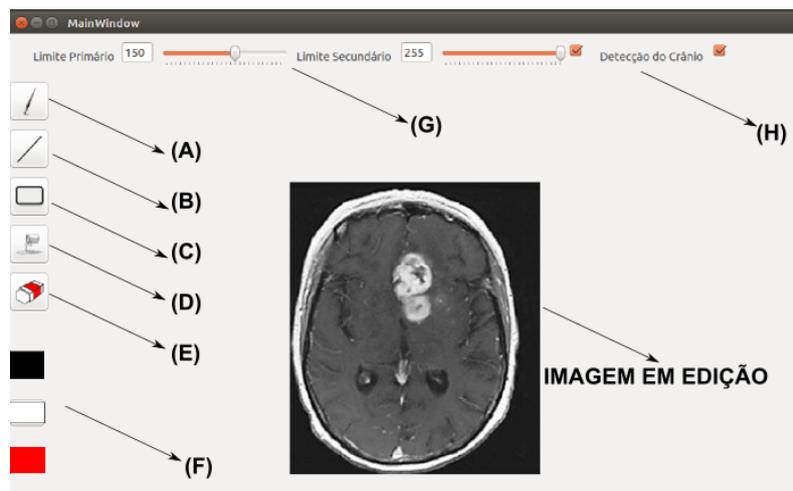


Figura 3. Funcionalidade do editor de imagens do GBM: (A) marcação livre, (B) marcação por junção de segmentos de retas (linhas), (C) marcação de uma área por retângulo, (D) preenchimento por cor, (E) apagar região selecionada (F) Paleta de cores disponíveis. Acima, há duas barras horizontais e duas caixas de marcação, (G) valor do limite para o primeiro e o segundo limiar (opcional) e (H) detecção automática do crânio (opcional).

Desse conjunto de funcionalidades, as mais importantes são:

- *A*, *B* e *C* que permitem a seleção da região do GBM, onde a opção *C* viabiliza a seleção da subimagem que contém a região de interesse, diminuindo o tamanho do arquivo final;
- A funcionalidade *G* parametriza a operação de limiarização (*thresholding*) que seleciona devidamente os pixels da região de interesse selecionada e binariza a imagem de saída.

As demais funções servem como opções para otimização da tarefa de marcação, por exemplo, o refazimento de alguma ação (apagando erros) e/ou visualização de estruturas (determinação da região do crânio). Sendo assim, o fluxo mais usual para a determinação do arquivo de saída e metadados para a inicialização da simulação é exibido na Figura 4.

A saída do programa demarcador gera dois arquivos:

- a imagem juntamente com todas as demarcações feitas antes da aplicação do limiar (*thresholding*),

- um segundo arquivo contendo informações a respeito da região retangular demarcada na imagem, os dados salvos são: o caminho absoluto da imagem, as coordenadas que representam os pontos iniciais e finais do retângulo e uma matriz onde o valor de cada elemento corresponde a cada uma das cores processadas pelo programa (preto é mapeado para 0, branco para 1 e vermelho para 2).

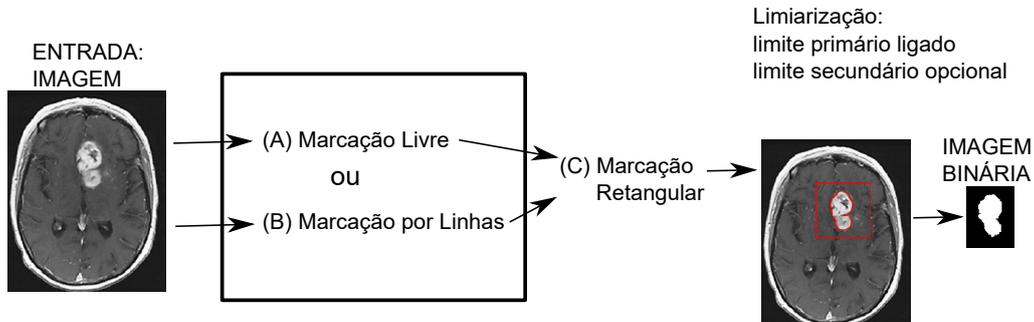


Figura 4. Fluxo usual para a determinação da região alvo de interesse.

5. Ambiente de Desenvolvimento

Nessa seção são descritos os recursos de programação utilizados para a feitura do programa aqui apresentado. Foi dada preferência a recursos disponíveis gratuitamente (Opensource Software). Os códigos do programa encontram-se disponíveis no seguinte URL: <https://github.com/Zuchis/imgEditor>

- Linguagem de programação: foi escolhido o Python como base para a construção do programa. Python é uma linguagem de programação de alto nível, interpretada, contempla os três paradigmas de programação e é de tipagem dinâmica, características que a torna uma linguagem extremamente flexível e de fácil manejo. Por ter uma boa legibilidade e uma vasta quantidade de módulos e frameworks, a adição de uma possível nova ferramenta ou método é uma tarefa fácil. Por esses motivos Python foi a linguagem escolhida para o desenvolvimento da aplicação. Outras bibliotecas não nativas ao Python também foram utilizadas, são elas: numpy, Python image library e pyQT.
- Numpy: biblioteca (Python) para realizar computação científica com o Python, isso inclui métodos numéricos, arrays e matrizes, foi utilizada no programa para auxiliar na aplicação do filtro gaussiano na função de detecção do crânio.
- Python Image Library (PIL): biblioteca (Python) utilizada para abrir, manipular e salvar imagens em diversos formatos. Os procedimentos disponíveis para manipulação de imagens incluem: operações pixelapíxel, aplicação de filtro e de máscara, manipulação de transparência, segmentação de imagens, entre outros.
- PyQT: vinculação para Python do famoso kit de ferramentas de interface gráfica QT, que fornece todos os elementos necessários para a interação com o usuário, como visualização da barra de ferramentas, detecção de inputs do mouse e do teclado e a posição do mouse na tela, visualização da imagem escolhida para demarcação e as alterações nela feita, disponibilização de um menu e caixas de diálogo para o usuário em caso de erros.

6. Resultados

A marcação de imagens foi aplicada, alguns exemplos são exibidos na Figura 5. Nessa figura, cada caso apresenta um par de imagens pré-tratamento e pós-tratamento (I_1, I_2), sem a marcação; bem como o par (I_1^i, I_2^o) apresentando a respectiva marcação das regiões de GBM, superpostas em vermelho sobre as imagens originais. A demarcação das regiões foi supervisionada por especialista médico e foi realizada por meio do programa aqui apresentado. O resultado é avaliado de maneira qualitativa pelo especialista e foi considerado eficaz na funcionalidade e usabilidade.

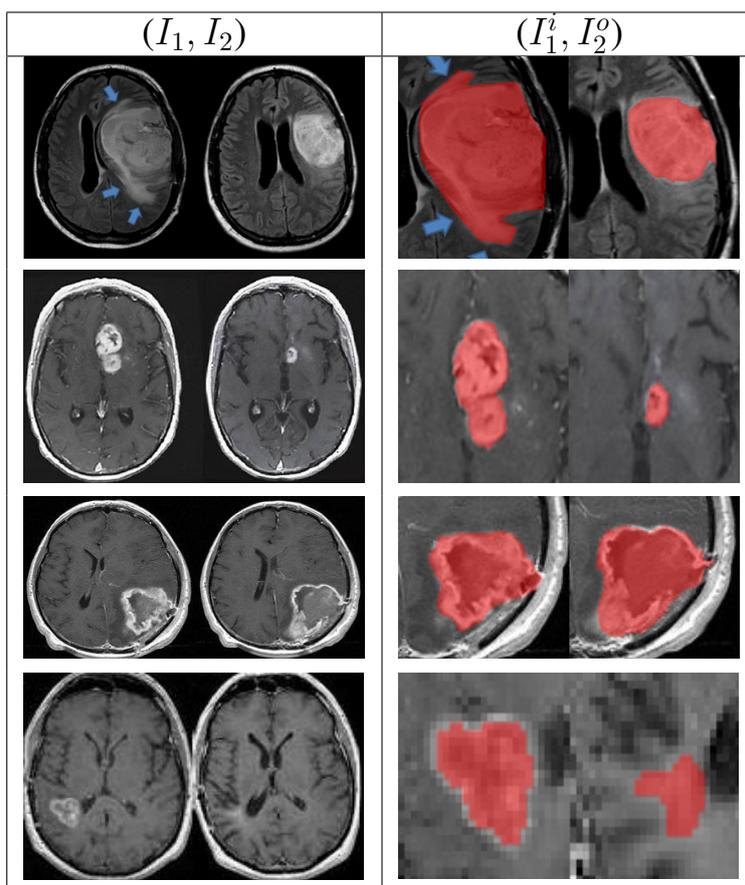


Figura 5. Exemplos de casos testados. Na coluna esquerda encontram-se os pares contendo imagens pré-tratamento I_1 e pós-tratamento I_2 . Por sua vez, na coluna da direita encontra-se a imagem I_1^i , a qual exibe a respectiva região de interesse (GBM) destacada em vermelho por meio do uso do programa demarcador, e a imagem I_2^o que exibe o resultado da simulação da evolução do tratamento, também em vermelho.

7. Conclusão

Este artigo descreveu um programa de computador voltado a marcação de regiões de GBM sobre imagens utilizando técnicas de processamento gráfico e de imagens. A funcionalidade desse software foi apresentada, bem como os fundamentos teóricos utilizados na sua implementação. Foram apresentados exemplos de utilização sobre imagens reais. Esses casos (anônimos) estavam representados em imagens selecionadas (por especialista médico) de tomogramas. Foi demonstrado o uso do programa de demarcação e relatada

a avaliação do especialista que supervisionou a implementação e uso do programa. Esta avaliação foi positiva, indicando o potencial do software. Pretende-se, em trabalhos futuros, dar continuidade ao desenvolvimento do programa acrescentando novas funções e processamentos de apoio à demarcação em questão.

8. Agradecimentos

Os autores agradecem o suporte da Universidade do Estado de Santa Catarina (UDESC - CCT - Joinville-SC) na forma de bolsa de Iniciação Científica no escopo do projeto “Aplicação de Autômato Celular no Estudo de Tumor Cerebral Glioblastoma Multifforme” e são gratos ao Dr. Edson Sydney de Campos.

Referências

- Friedman, H. S. et al. (2009). Bevacizumab alone and in combination with irinotecan in recurrent glioblastoma. *Journal of Clinical Oncology*, 27(28):4733–4740.
- Grossman, S. A. et al. (2010). Survival of patients with newly diagnosed glioblastoma treated with radiation and temozolomide in research studies in the United States. *Clinical Cancer Research*, 16(8):2443–2449.
- Guedes, A. F. B. D. (2010). Estudo de marcadores de diagnóstico e prognóstico em gliomas. Master’s thesis, Universidade de Trás-os-Montes e Alto Douro Vila Real.
- Holland, E. C. (2000). Glioblastoma multifforme: the terminator. *Proceedings of the National Academy of Sciences*, 97(12):6242–6244.
- Kari, J. (2005). Theory of cellular automata: A survey. *Theoretical Computer Science*, 334(1):3–33.
- Monteagudo, Á. e Santos, J. (2014). Studying the capability of different cancer hallmarks to initiate tumor growth using a cellular automaton simulation. application in a cancer stem cell context. *Biosystems*, 115:46–58.
- Oliveira, G. (2003). Autômatos celulares: aspectos dinâmicos e computacionais. *III Jornada de Mini-cursos em Inteligência Artificial (MCIA)*, 8:297–345.
- Pedriani, H. e Schwartz, W. R. (2008). *Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações*. Editora Thomson Learning.
- Pinheiro, L. P. (2015). Aplicação de autômato celular para predição de resposta do glioblastoma multifforme ao tratamento com bevacizumabe. Trabalho de Conclusão de Curso, Universidade do Estado de Santa Catarina (Joinville-SC-Brasil).
- Ruuth, S. J. (1995). Implicit-explicit methods for reaction-diffusion problems in pattern formation. *Journal of Mathematical Biology*, 34(2):148–176.
- Wolfram, S. (1984). Cellular automata as models of complexity. *Nature*, 311(5985):419–424.