

Testes de Validação do Algoritmo de  
Reformatação Curvilínea Implementado no  
VMTK-Neuro 3.1

A.V.S. Lima, W. Loos, S.-T. Wu

26 de fevereiro de 2019

Meus agradecimentos à Profa. Ting pela orientação e introdução a um tema de pesquisa que não tinha conhecimento, ao doutorando Wallace pelo empenho dedicado ao desenvolvimento deste trabalho, aos membros da equipe de pesquisa pela pronta disposição em esclarecer as minhas dúvidas, aos colegas do LCA (Laboratório de Engenharia de Computação e Automação Industrial) da FEEC pelo companheirismo, e ao SAE (Serviço de Apoio ao Estudante) pelo suporte financeiro.

# 1 Introdução

O VMTK-Neuro (*Visual Manipulation Toolkit for Neuroimages*) é um aplicativo de exploração de neuroimagens desenvolvido por uma equipe de pesquisadores em Análise Visual de Dados Médicos da Faculdade de Engenharia Elétrica e de Computação da Unicamp[?], com a finalidade de promover uma forma mais intuitiva para explorar o cérebro. Este aplicativo possui ferramentas que permitem a um especialista manipular imagens de ressonância magnética, representadas no formato DICOM (*Digital Imaging and Communications in Medicine*), através das visualizações coordenadas em 2D e 3D. Entre as diversas ferramentas desenvolvidas pela equipe está a reformatação curvilínea.

Reformatação curvilínea é uma técnica que tem como finalidade realizar cortes curvilíneos de um cérebro. Ela complementa as reformatações multiplanares na localização de displasia cortical focal, causa comumente associada à epilepsia refratária [?]. O grande diferencial da técnica proposta por Loos *et al.*[?] em relação às técnicas existentes é a preservação dos tecidos extra-cerebrais (vasos superficiais) e o espaço de referência ser o nativo do paciente, viabilizando também a sua aplicação em planejamento neurocirúrgico.

A técnica de reformatação mencionada consiste na estimativa de uma malha triangular aderente ao escalpo, através de uma sequência de deformações geométricas (Figura 1). Malhas triangulares são aplicadas para relacionar as amostras discretas dos volumes escaneados com as imagens visualizadas. Os vértices dessa malha, inicialmente esférica, são deslocados na direção oposta do vetor normal ao vértice de cada triângulo até a superfície do cérebro. Esta malha ajustada ao cérebro é utilizada para fatiar o cérebro curvilinearmente.

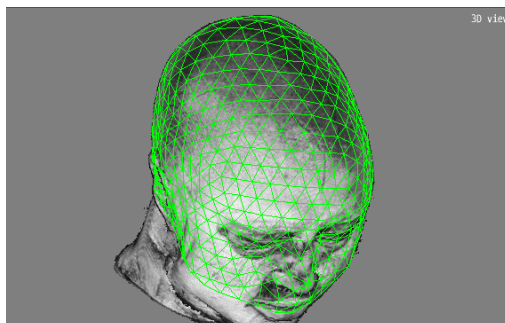


Figura 1: Malha, em verde, gerada numa reformatação curvilínea

Para compreender o procedimento de deformação da malha, é importante destacar algumas regiões do crânio e cérebro que podem ser observadas na Figura[2]. Sabendo que o espaço subaracnóideo (entre as meninges aracnóide e pia-máter) é preenchido pelo líquido cefalorraquidiano (LCR), a equipe conjecturou que este líquido apresenta a menor intensidade entre os sinais de ressonância magnética ponderada em T1 e que este fato pode ser utilizado para guiar o deslocamento dos vértices do escalpo (superfície craniana) para a superfície cerebral.

No decorrer deste trabalho levantou-se curvas para analisar as variações dos sinais de ressonância magnética entre o escalpo e a substância branca do cérebro, para certificar se de fato a intensidade do sinal do LCR é a menor em exames de ressonância magnética ponderada em T1. Conduzimos ainda testes num

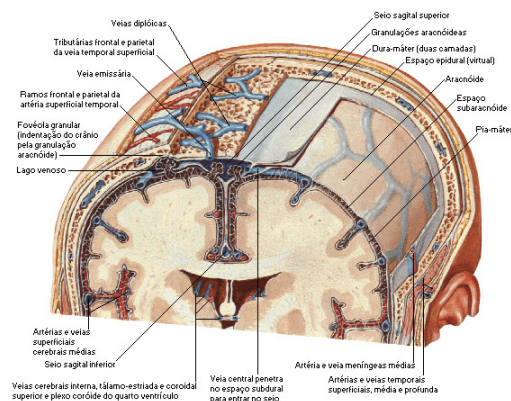


Figura 2: Estrutura anatômica do cérebro. Fonte: NETTER, Frank H.. Atlas de Anatomia Humana. 2ed. Porto Alegre: Artmed, 2000.

conjunto de 676 volumes para avaliar a qualidade dos cortes obtidos em termos da exposição da superfície cortical e aprimorar a ferramenta de reformatação curvilínea. Foi também proposto um algoritmo para a reconstrução da borda da malha de corte.

Este relatório está organizado em seguintes seções: materiais e métodos, resultados, conclusão e apêndice. Na Seção 2 encontram-se a descrição das imagens utilizadas para os testes, as categorizações feitas entre essas imagens, a forma e os critérios como os testes foram aplicados e a proposta de um algoritmo alternativo ao utilizado para a construção de uma curva para a borda da malha de corte. Na Seção 3 encontram-se os gráficos de profundidade x intensidade, estatísticas sobre o comportamento dos diferentes grupos de volumes com a aplicação das diferentes versões da reformatação e algumas considerações sobre estes dados e sobre a geometria da borda da malha de corte. Na Seção 4 encontram-se as observações finais do trabalho desenvolvido. Por fim, no apêndice encontram-se pseudo-códigos do algoritmo de reconstrução da curva de borda da malha de corte.

## 2 Materiais e Métodos

De modo geral, os testes consistiram na aplicação da reformatação curvilínea em imagens de ressonância magnética ponderadas em T1, sem e com adição de contraste de gadolínio. Estas imagens encontram-se em um repositório de imagens médicas hospedado no Laboratório de Neuro-Imagens (LNI) da Faculdade de Ciências Médicas da Unicamp (FCM) e foram cedidas pela Professora Doutora Clarissa Lin Yassuda.

### 2.1 Sinais ao Longo de Raios Lançados a partir do Escalpo

A forma de captação do padrão de variação das intensidades dos sinais foi por meio da utilização do VMTK-Neuro: amostrar as intensidades dos *voxels* existentes em um raio de comprimento  $R$  definido pelo usuário, que se prolonga a partir de um vértice da malha triangular, selecionado por meio da ferramenta

*paint* no VMTK-Neuro. Além do mais, esse raio de comprimento  $R$  se prolonga na direção oposta ao vetor normal ao vértice selecionado. A partir das intensidades dos *voxels* com que o raio intersecciona, foi construída uma curva de intensidade x profundidade, segundo a qual é determinado e calculado um ponto de mínimo da curva a fim de avaliar se este mínimo corresponde a LCR, como proposto no trabalho de Loos *et al.* [?], onde a malha deve parar.

Especificamente, para os nossos testes de validação, foi demarcada com a ferramenta *paint* uma região no escalpo que contém o vértice a partir do qual queremos prolongar um raio para a amostragem de intensidades. Em cada volume, foram selecionados quatro vértices distribuídos sobre o lóbulo frontal (Figura 3), quatro sobre o lóbulo parietal, quatro sobre o lóbulo occipital e dois sobre cada um dos lados do lóbulo temporal. Projetou-se um raio de 60mm de comprimento em cada vértice selecionado para certificar-se visualmente a correspondência entre o LCR e o mínimo das intensidades. A certificação, em si, dessa correspondência não foi feita neste trabalho .

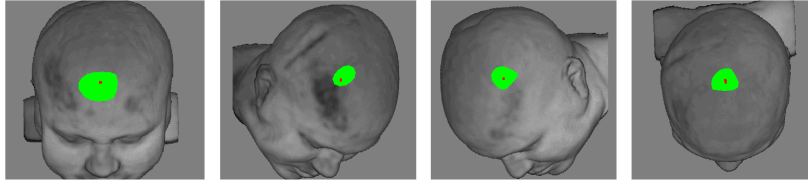


Figura 3: Amostragem dos quatro vértices do lobo frontal.

## 2.2 Qualidade da Reformatação Curvilínea

Como suporte ao aprimoramento do algoritmo de reformatação curvilínea em desenvolvimento no laboratório LCA, quatro versões deste algoritmo foram avaliadas em relação à sua qualidade de corte. Na primeira versão, o algoritmo ajusta as malhas de corte à superfície cortical, exceto no lobo temporal. Na segunda versão, o algoritmo é construído em cima da técnica de *skull stripping*, portanto, o ajuste é feito sobre a superfície cortical. Na terceira versão, o algoritmo ajusta as malhas de corte à superfície cortical incluindo o lobo temporal. E, finalmente, na quarta versão foi ainda adicionada ao algoritmo da terceira versão um recuo de 7mm na direção do escalpo.

Para análise da qualidade de corte, foi utilizada uma curva que mostra nos planos sagital, coronal e axial, em que lugar do espaço do volume e de que forma o corte ocorreu. Esta curva, já disponível no VMTK-Neuro, é, de fato, a intersecção da malha triangular com estes planos.

### 2.2.1 Primeira Versão: ajustes ao escalpo

Aqui aplicou-se a reformatação curvilínea a um conjunto de 72 volumes RM-T1. Estes volumes foram separados, então, por gênero e idade, nas faixas 10-20, 21-30, 31-40, 41-50, 51-60, 61-70, 71-80, 81-90, com geralmente 5 pacientes por faixa etária e por sexo para se analisar a qualidade do corte da reformatação curvilínea.

### 2.2.2 Segunda Versão: *Skull Stripping*

A técnica de reformatação apresentada na Seção 1 passa por um ajuste por meio da utilização de um algoritmo de *skull stripping*. O volume, após o procedimento de *skull stripping*, foi utilizado como uma máscara para orientar os vértices da malha de corte à superfície do lobo temporal.

Foi elaborado um estudo nos volumes (RM-T1) de 676 pacientes de controle separados por gênero e faixa etária, as mesmas divisões do método explicitado na Seção 2.2.1. O estudo consistiu em avaliar se ajustando a malha a esta máscara os lobos temporais ficariam melhor expostos, classificando os resultados em três grupos:

1. Não expor nenhum lobo (Há problemas em expor todos os lobos)
2. Expor exclusivamente o temporal (Há problema em expor os demais lobos)
3. Não expor exclusivamente o lobo temporal (Há problemas em expor somente o temporal)

### 2.2.3 Terceira Versão: Ajuste fino ao lobo temporal

Para esta nova versão, no que se refere a prática da reformatação, o método de aplicação da reformatação mudou apenas em relação ao acréscimo de pontos marcados nas extremidades dos olhos e de uma medida manual da espessura do escalpo para guiar a malha para dentro do LCR [?]. O usuário precisa marcar um ponto em cada epicanto externo, para que o algoritmo encontre o lóbulo temporal.

Foram utilizadas as imagens de ressonância magnética com contraste para fundamentar o potencial de aplicação em planejamento cirúrgico: os volumes quando obtidos com contraste de gadolínio mostram os vasos sanguíneos.

A reformatação foi aplicada novamente em dois conjuntos de volumes:

1. Em 676 volumes RM-T1 separados por gênero e faixas etárias de 10-20, 21-30, 31-40, 41-50, 51-60, 61-70, 71-80, 81-90.
2. Em um conjunto de 6 volumes com adição de contraste de gadolínio e supressão de gordura (GAD-T1wMR1).

Buscou-se com esse teste validar se houve melhora na qualidade do corte, quando comparado com a versão anterior (Seção 2.2.2) e também atestar a reformatação em volumes obtidos com contraste.

### 2.2.4 Quarta Versão: Ajuste com Recuo da Malha de Corte

Nesta última avaliação, mais um aprimoramento foi feito para evitar remoção parcial do córtex: o algoritmo estima a espessura do escalpo à pia-máter sobre os lobos frontal e parietal para guiar a malha para dentro do LCR, e aplica um recuo na malha de 7mm, partindo da profundidade da espessura calculada na direção do vetor normal a superfície da malha.

Nesta fase aplicamos a reformatação a 168 volumes escolhidos aleatoriamente dentre aqueles 676 volumes de controle em três faixas etárias, 21–30, 31–40 e 41–50, sendo metade do gênero masculino, metade do gênero feminino. Buscou-se atestar se a reformatação teria mantido a qualidade do corte conseguida na versão exposta na Seção 2.2.3.

### 2.3 Aprimoramento da Geometria da Intersecção das Fatias com a Malha de Corte

Ao longo do desenvolvimento do algoritmo que ajusta malhas às meninges, percebeu-se que a intersecção das fatias com a malha não é sempre computada corretamente, como na extremidade inferior do lobo occipital ilustrada na Figura 4. Já mais familiarizado com o ambiente de desenvolvimento, assumi a tarefa de melhorar a geometria dessa borda.

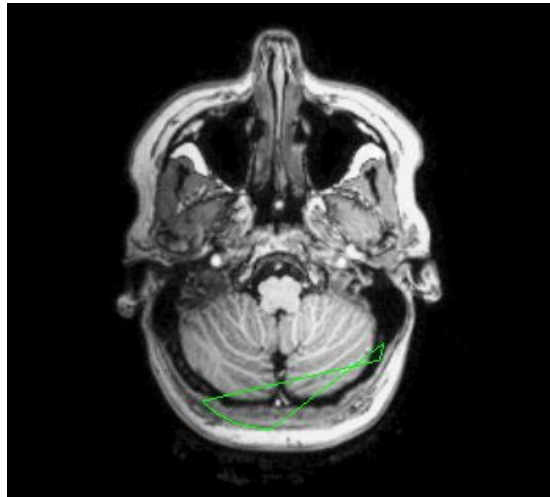


Figura 4: Vista axial da curva deformada.

A malha utilizada para os cortes é composta por triângulos, como ilustrado pela Figura 1. E estes triângulos não são paralelos aos planos sagital, coronal e occipital. Baseando-se no fato de que dois triângulos adjacentes dessa malha, quando interseccionados em suas arestas por um plano possuem um ponto em comum tem-se uma sequência de pontos, dois por triângulo, relacionados entre si por um ponto proveniente de uma aresta em comum aos dois triângulos.

Utilizamos este fato para ordenar dinamicamente os pontos de intersecção dos triângulos com o plano, de forma que, ao concluir o cômputo das intersecções com todos os triângulos, temos uma sequência de pontos que corresponde à curva de intersecção: par a par, os novos pontos computados são comparados com os pontos já ordenados para sequenciá-los. Um pseudo-código do algoritmo proposto está descrito na Seção 5.

## 3 Resultados

Nesta seção é apresentada uma seleção de resultados obtidos.

### 3.1 Curvas dos Sinais ao longo de Raios Lançados

Devido à vasta quantidade de imagens produzidas nesta análise, optou-se por mostrar na Figura 5 somente as curvas relativas aos lóbulos de um volume JNB arbitrariamente escolhido.

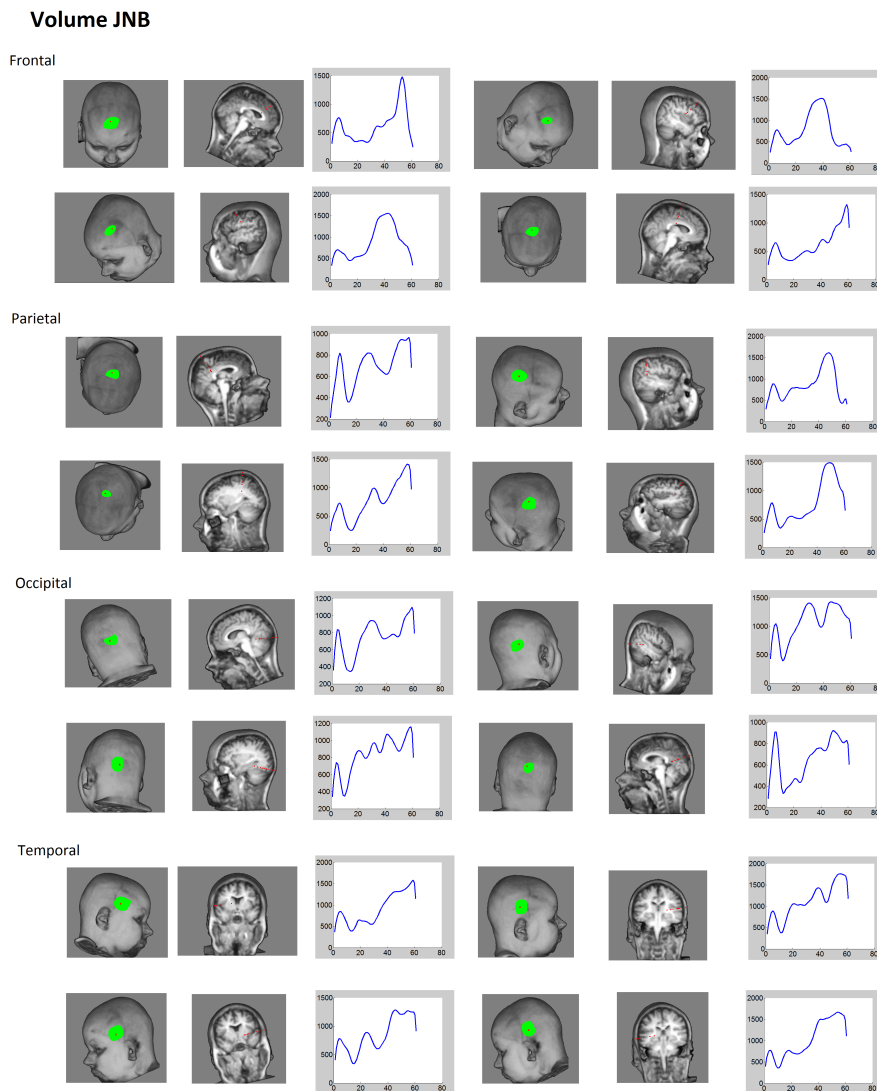


Figura 5: Curvas intensidade x profundidade do vértice prolongado de um volume JNB

### 3.2 Qualidade do Corte em Reformatação Curvilínea

Buscou-se avaliar visualmente se os lóbulos temporal, parietal, occipital e frontal poderiam ser visualizados e, com o auxílio da interseção da malha de corte, afirmar se houve de fato remoção indevida de tecidos.

#### 3.2.1 Primeira Versão

A reformatação proporcionou a visualização dos lóbulos temporal, parietal, occipital e frontal em 58% dos casos, os volumes reformatados podem ser visua-



lizados na Figura 6.

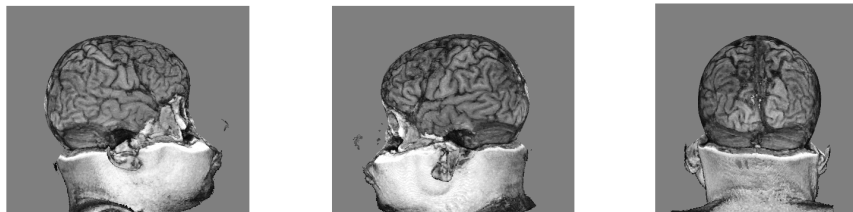


Figura 6: Visualização da superfície cortical

A relação de volumes que passaram no teste com as faixas etárias e sexo podem ser visualizadas na Figura 7. Cada faixa etária possui em média 5 pacientes. Os pacientes que passaram pelo teste sem nenhuma distinção por sexo ou idade, são de um número de 8, pois em alguns casos não foi possível identificar sexo e idade pelos arquivos.

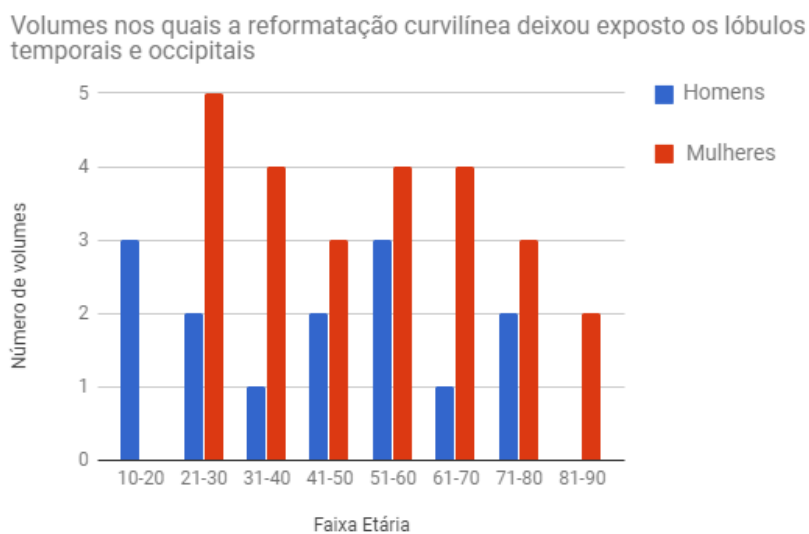


Figura 7: Resultados do teste de corte com a primeira versão do algoritmo de reformatação curvilínea.

Nesse primeiro momento notou-se uma variação no corte relacionada ao sexo, mas não à idade. Aparentemente, o algoritmo funcionou melhor em volumes provenientes de pessoas do sexo feminino.

Notou-se ainda que a variação depende da estrutura dos tecidos sobre os lobos. Onde o LCR esteja melhor definido, por exemplo distribuído em uma faixa mais larga, o corte foi mais preciso. Neste sentido a região do lobo temporal merece destaque, uma vez que sobre este lobo, o LCR é menos distinguível nas

imagens. Isso acontece porque o lobo temporal fica abaixo da fossa temporal circundada pelo arco zigomático. O espaço entre a fossa temporal e o arco zigomático é preenchido com a massa óssea, músculos e gordura, que faz com que esta região do cérebro fique mais funda em comparação a outras regiões. Portanto em comparação com a formação craniana das demais regiões e tecidos que sobrepõem a superfície cortical das demais regiões, essa distribuição específica de tecidos resulta em uma distribuição específica de intensidades na imagem e dificulta para um algoritmo definir onde está o mínimo global dessas intensidades, como demonstra o caso ilustrado na Figura 8. A figura mostra o volume de um paciente do sexo feminino e na faixa etária de 31-40 anos, na qual após a reformatação curvilínea o lobo temporal ficou ainda parcialmente coberto nas imagens da esquerda e central.



Figura 8: Resultado de corte mal sucedido da primeira versão do algoritmo.

### 3.2.2 Segunda Versão: *Skull Stripping*

Feita a avaliação como explicitada no início da Seção 3.2 e classificando os volumes avaliados segundo os marcadores definidos na Seção 2.2.2, calculou-se que 33% dos pacientes apresentavam os marcadores responsáveis por sinalizar se o lobo temporal não era exposto, ou seja, marcadores 1 e 3.

Com a utilização desta técnica, houve uma melhora considerável nos casos em que todos os lobos foram expostos. Entretanto, o tempo de execução do corte aumentou, devido a forma como a reformatação dialoga com a máscara. Também devido à versão do algoritmo de *skull stripping* utilizada que não era otimizada. O que foi decisivo no abandono desta versão foi, no entanto, remoção dos detalhes da superfície cortical.

### 3.2.3 Terceira Versão

Com esta versão apenas 5 volumes não tiveram o lobo temporal exposto, devido a baixa resolução das imagens de RM, como ilustra a Figura 9.

Esta versão, permitiu também a visualização de todos os lobos dos volumes contrastados com gadolínio.

### 3.2.4 Quarta Versão

Todos os volumes que passaram por essa avaliação tiveram todos os lobos expostos. A figura 10 ilustra onde aconteceram os cortes. Estes estão dentro do LCR. Não estão nem dentro do córtex cerebral, retirando informação da

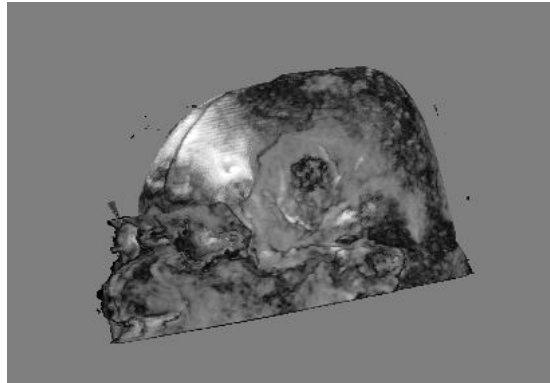
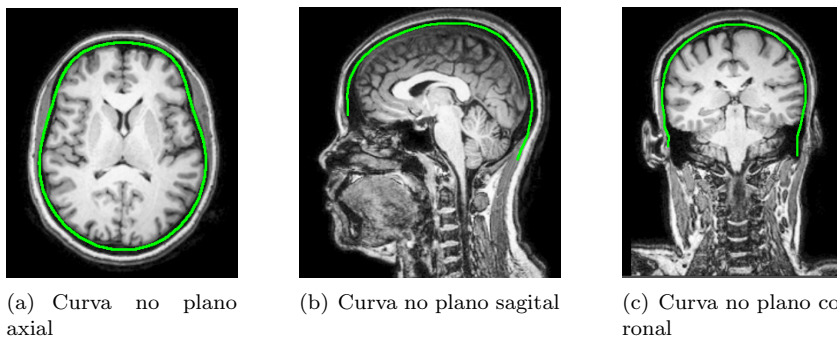


Figura 9: RM de baixa resolução

superfície cortical, nem dentro do crânio, ocultando informações da superfície do cortical.



(a) Curva no plano axial

(b) Curva no plano sagital

(c) Curva no plano coronal

Figura 10: Aqui tem-se um exemplo de como a curva nos planos 2D auxiliou na avaliação dos cortes

### 3.2.5 Considerações Sobre os Resultados dos Testes da Terceira e Quarta Versões

Para o usuário é mais simples realizar a reformatação na segunda versão do que na primeira, uma vez que não é necessário que este realize medições da espessura do escalpo.

Nestas avaliações ficou evidente que a reformatação teve o rendimento buscado, principalmente ao que se refere a exposição do lobo temporal, que era uma das grandes dificuldades a ser superadas e que fora superada segundo as avaliações da terceira e quarta versões. Apenas cinco volumes tiveram algum tipo de obstrução do lobo temporal na terceira versão e na quarta versão todos os lobos foram expostos em todas as imagens.

A qualidade dos cortes realizados nos volumes de ressonância magnética com contraste de gadolínio mostraram-se suficientes para o propósito de expor os vasos sanguíneos na superfície cortical. Todos os volumes submetidos a reformatação tiveram a superfície dos lóbulos exposta [?].

### 3.3 Geometria da Borda da Malha de Corte

O algoritmo criado ordena alguns conjuntos de pontos como demonstrado na Figura 11 (b), mas em outros não, como na Figura 11 (a).

Identificou-se até o momento que o que causa a não ordenação é a falta de algum ponto ou par de pontos de triângulos adjacentes.

Para contornar o problema tentou-se melhorar a precisão dos pontos flutuantes, que são os pontos utilizados para o cálculo dos pontos de intersecção.

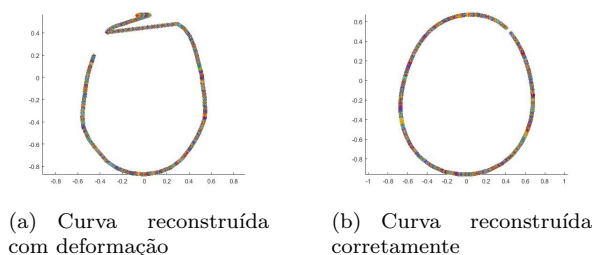


Figura 11: Reconstruções das bordas da malha de corte

## 4 Conclusão

Tendo em vista o intento deste trabalho, exposto na Seção [1], vê-se que referente ao levantamento das curvas de intensidade x profundidade, estas possuem ponto de mínimo global e cabe a outro trabalho a constatação de que este ponto de mínimo é realmente coincidente ao LCR.

Sobre o algoritmo para a organização dos pontos de intersecção da malha de corte, verificou-se que para que seja eficaz, o método proposto precisa passar por aperfeiçoamentos para que possa reconstruir fielmente a borda.

Por último, diferentes versões do algoritmo de reformatação curvilínea foram testadas, até que se conseguisse uma versão que expôs todos os lobos e não retirou informações da superfície cortical.

## 5 Apêndice

Neste apêndice é detalhado o algoritmo de ordenação de um par de pontos de intersecção,  $P_i$  e  $P_{i+1}$ , de cada triângulo de uma malha com o plano para obter uma sequência  $L_k$  da intersecção desta malha com o plano, conforme mostra o pseudocódigo 1. Na primeira chamada, cria-se sempre uma sub-lista  $L_k$  (linha 4) onde são inseridos os pontos  $P_i$  e  $P_{i+1}$ . Do contrário, é chamada a função LISTA NÃO VAZIA (linha 9), cujo pseudocódigo se encontra na seção 5.1.

O algoritmo foi implementado em C++.

---

**Algoritmo 1: ORDENA PONTOS**

---

**Entrada:** Par de pontos:  $P_i, P_{i+1}$ , lista  $L$

**Saída:** Lista  $L$  atualizada

```
1 início
2   para cada ponto  $P_i$  faça
3     se a lista  $L$  estiver vazia então
4       Cria uma sub-lista lista  $L_k$  para  $L$ 
5       insere  $P_i$  em  $L_k$ 
6       insere  $P_{i+1}$  em  $L_k$ 
7     fim
8   senão
9     LISTA NÃO VAZIA( $P_i, P_{i+1}, L$ )
10  fim
11 fim
12 fim
```

---

## 5.1 Lista Não Vazia

Esta função trata da inserção dos pontos  $P_i$  e  $P_{i+1}$  de um lado de um triângulo em sub-listas  $L_k$ . Como cada sub-lista corresponde a uma sequência de pontos conectados, esta inserção pode ser feita de três formas: quando um dos pontos já existe numa das sub-listas ou quando ambos pontos já estão inseridos em alguma sub-lista. Para tanto, dois marcadores  $I_A$  e  $I_B$  são inicializados (linha 2).

Então, busca-se por  $P_i$  e  $P_{i+1}$  nas  $l_k$  sub-listas (linha 3 e 4). Se  $P_i$  é encontrado em uma sub-lista armazenam-se os índices  $\{k, j\}$  onde  $j$  e  $k$  denotam respectivamente a posição dentro de uma sub-lista (uma sequência de segmentos ordenados) e a posição desta sub-lista na lista  $L$ , do ponto em  $I_A$  (linha 6). Caso  $P_{i+1}$  seja encontrado em uma sub-lista, os índices  $\{k, j\}$  correspondentes são armazenados em  $I_B$  (linha 9). Se não forem encontrados,  $I_A$  e  $I_B$  permanecem nulos.

Dessa forma, baseado nos estados possíveis de  $I_A$  e  $I_B$ , existirão quatro possibilidades de inserção do segmento  $P_i$  e  $P_{i+1}$ , que culminam em quatro condições:

1.  $I_A$  é nulo e  $I_B$  é nulo: cria-se uma nova sub-lista e insere  $P_i$  e  $P_{i+1}$  (linhas 13 à 17).
2.  $I_A$  é nulo e  $I_B$  é não nulo: chama-se a função INSERE  $P_i$  (linha 20).
3.  $I_A$  é não nulo e  $I_B$  é nulo: chama-se a função INSERE  $P_{i+1}$  (linha 23).
4.  $I_A$  é não nulo e  $I_B$  é não nulo: chama-se a função LIGA LISTAS (linha 26).

As funções INSERE e LIGA\_LISTAS são detalhadas nas seções 5.2 e 5.3,

respectivamente.

---

**Algoritmo 2:** LISTA NÃO VAZIA

---

**Entrada:** Pontos  $P_i$  e  $P_{i+1}$  e lista  $L$

**Saída:** Lista  $L$  atualizada

```
1 início
2    $I_A = I_B = \text{NULL}$ 
3   para cada elemento  $k$  contido na lista  $L$  faça
4     para cada elemento  $j$  contido na sub-lista  $L_k$  faça
5       se  $P_i == P_j$  então
6          $I_A$  recebe  $\{k, j\}$ 
7       fim
8       se  $P_{i+1} == P_j$  então
9          $I_B$  recebe  $\{k, j\}$ 
10      fim
11    fim
12  fim
13  se  $I_A$  e  $I_B$  são vazios então
14    cria uma nova sub-lista  $L_k$ 
15    insere  $P_i$  em  $L_k$ 
16    insere  $P_{i+1}$  em  $L_k$ 
17    insere  $L_k$  em  $L$ 
18  fim
19  se  $I_A$  é vazio e  $I_B$  é não vazio então
20    INSERE  $P_i$  ( $L, I_B, P_i$ )
21  fim
22  se  $I_A$  é não vazio e  $I_B$  é vazio então
23    INSERE  $P_{i+1}$  ( $L, I_A, P_{i+1}$ )
24  fim
25  se  $I_A$  e  $I_B$  são não vazios então
26    LIGA LISTAS ( $I_A, I_B$ )
27  fim
28 fim
```

---

## 5.2 Insere $P_i$ e $P_{i+1}$

As duas funções descritas pelos dois algoritmos seguintes são as encarregadas de inserir  $P_i$  e  $P_{i+1}$ . No primeiro caso (algoritmo: 3), é testado se  $I_B$  é o primeiro ou o último elemento da sub-lista (posição  $k$  da lista  $L$ ) que o contém. Caso  $I_B$  seja o último elemento da sub-lista,  $P_i$  é inserido no fim da sub-lista que contém  $I_B$  (linha 7). Se  $I_B$  for o primeiro elemento,  $P_i$  é inserido ao início da sub-lista (linha 3).

No segundo caso (algoritmo: 4), é testado se  $I_A$  é o primeiro ou o último elemento da sub-lista (novamente posição  $k$  da lista  $L$ ). Caso  $I_A$  seja o último elemento da sub-lista,  $P_{i+1}$  é inserido no fim da sub-lista que contém  $I_A$  (linha 7). Se  $I_B$  for o primeiro elemento,  $P_{i+1}$  é inserido ao início da sub-lista (linha

3).

---

**Algoritmo 3: INSERE  $P_i$** 

---

**Entrada:**  $L, I_B, P_i$

**Saída:** Lista  $L$  com  $P_i$  atualizada

```
1 início
2   se  $I_B ==$  primeiro elemento da sub-lista  $L_k$  que o contém então
3     | insere  $P_i$  no início de  $L_k$ 
4   fim
5   senão
6     | se  $I_B ==$  último elemento da sub-lista  $L_k$  que o contém então
7       | insere  $P_i$  no fim de  $L_k$ 
8     fim
9   fim
10 fim
```

---

---

**Algoritmo 4: INSERE  $P_{i+1}$** 

---

**Entrada:**  $L, I_A, P_{i+1}$

**Saída:** Lista  $L$  com  $P_{i+1}$  atualizada

```
1 início
2   se  $I_A ==$  primeiro elemento da sub-lista  $L_k$  que o contém então
3     | insere  $P_{i+1}$  no início de  $L_k$ 
4   fim
5   senão
6     | se  $I_A ==$  último elemento da sub-lista  $L_k$  que o contém então
7       | insere  $P_{i+1}$  no fim de  $L_k$ 
8     fim
9   fim
10 fim
```

---

### 5.3 Liga Sub-listas

Esta função se refere à ação a ser executada quando ambos  $I_A$  e  $I_B$  são encontrados. Neste caso temos quatro alternativas:

1. Se  $I_A$  e  $I_B$  estão na primeira posição das sub-listas que os contém, então inverte-se a sub-lista que contém  $I_B$  e insere-se esta sub-lista ao início da sub-lista que contém  $I_A$  (linha 4).
2. Se  $I_A$  e  $I_B$  estão na última posição das sub-listas que os contém, então inverte-se a sub-lista que contém  $I_B$  e insere-se esta sub-lista ao fim da sub-lista que contém  $I_A$  (linha 8).
3. Se  $I_A$  está na primeira posição da sub-lista que o contém e  $I_B$  está na última posição da sub-lista que o contém, então insere-se a sub-lista que contém  $I_B$  ao início da sub-lista que contém  $I_A$  (linha 11).
4. Se  $I_A$  está na última posição da sub-lista que o contém e  $I_B$  está na primeira posição da sub-lista que o contém, então insere-se a sub-lista que contém  $I_B$  ao fim da sub-lista que contém  $I_A$  (linha 14).

---

**Algoritmo 5:** LIGA SUB-LISTAS

---

**Entrada:**  $I_A, I_B$ **Saída:** Sub-listas  $L_k$  ligadas e ordenadas

```
1 início
2   se  $I_A ==$  primeiro elemento da sub-lista e  $I_B ==$  primeiro elemento
   da sub-lista então
3     |   inverte a sub-ista de  $I_B$ 
4     |   insere a lista invertida no início da lista de  $I_A$ 
5   fim
6   se  $I_A ==$  último elemento da sub-lista e  $I_B ==$  último elemento da
   sub-lista então
7     |   inverte a sub-ista de  $I_B$ 
8     |   insere a lista invertida no fim da lista de  $I_A$ 
9   fim
10  se  $I_A ==$  primeiro elemento da sub-lista e  $I_B ==$  último elemento da
   sub-lista então
11    |   insere a sub-lista de  $I_B$  no início da lista de  $I_A$ 
12  fim
13  se  $I_A ==$  último elemento da sub-lista e  $I_B ==$  primeiro elemento da
   sub-lista então
14    |   insere a sub-lista de  $I_B$  no fim da lista de  $I_A$ 
15  fim
16 fim
```

---