

**Universidade Estadual de Campinas - UNICAMP**

---

**Instituto de Computação - IC**



## **Visualização de Imagem Volumétrica MO815B**

**Relatório Final**

**Marianna Novaes Martins - 311615  
Rodolfo Luis Tonoli - 105652**

**Docente: Alexandre X. Falcão**

**Campinas**  
06 de dezembro de 2016

## Índice

Introdução.....	3
Desenvolvimento.....	4
Resultados.....	12
Conclusão.....	23
Referências.....	24

## Introdução

Visualização volumétrica é o conjunto de técnicas utilizadas na visualização de dados relacionada com a representação, manipulação e visualização de conjuntos de dados multidimensionais (geralmente de três dimensões), tendo como principal objetivo o processo de projetar um dado desse conjunto em uma imagem de duas dimensões (2D). Este processo é feito através da utilização de técnicas de computação gráfica e processamento de imagens.

Este trabalho aborda métodos para visualização de imagens volumétricas e apresenta as diferentes formas de explorar e exibir tais dados, manipulando parâmetros geométricos e visuais. Assim, este relatório descreve os algoritmos que foram desenvolvidos na disciplina de Visualização de Imagem Volumétrica, ministrada pelo professor Alexandre Falcão.

Os métodos e técnicas implementadas foram divididas nas seguintes tarefas:

- Tarefa 1: Exibir imagens 2D (cortes) dos planos axial, coronal ou sagital de uma imagem médica 3D, com liberdade para adequar-se à forma preferencial de visualização de profissionais de diferentes áreas da saúde.
- Tarefa 2: Realçar brilho e contraste dos cortes obtidos na tarefa anterior.
- Tarefa 3: Sobrepor objetos resultantes da segmentação da imagem 3D no corte obtido da primeira tarefa utilizando cor e transparência.
- Tarefa 4: Desenhar retas ligando os vértices de uma imagem 3D rotacionada, caso estejam visíveis.
- Tarefa 5: O caso geral da primeira tarefa, exibir um corte planar centralizado em um ponto arbitrário de uma imagem 3D de acordo com um vetor de visualização.
- Tarefa 6: Reformatação de um segmento de reta repetindo a tarefa anterior ao longo de pontos dessa reta, gerando um novo conjunto de dados tridimensional.
- Tarefa 7: Implementação da técnica projeção de intensidade máxima ou MIP, em inglês.
- Tarefa 8: Rendering de superfícies, aplicando o modelo de iluminação de Phong.

## Desenvolvimento

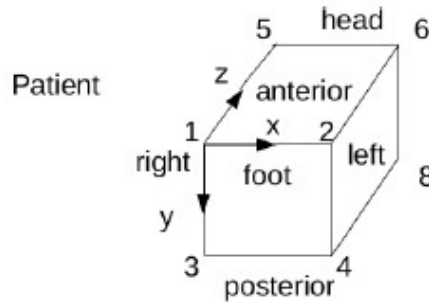
Os trabalhos foram desenvolvidos na linguagem C, tendo como base a biblioteca de funções (conhecida como libmc920) disponibilizada pelo professor da disciplina.

As três primeiras atividades foram realizadas em sistemas operacionais diferentes, Linux e Windows.

### • Tarefa 1

Escreva uma função C para extrair um corte (imagem 2D) em um dado eixo  $A \in \{X, Y, Z\}$  de uma imagem 3D, selecionando uma dada ordem entre os eixos  $B \in \{X, Y, Z\} \setminus \{A, C\}$  e  $C \in \{X, Y, Z\} \setminus \{A, B\}$ , e uma entre as quatro possíveis coordenadas de origem.

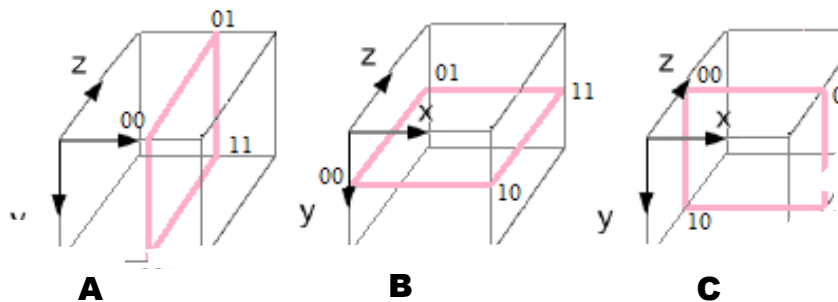
A posição do paciente na aquisição da imagem, Figura 1, foi essencial para identificar a forma de visualização de uma imagem 2D.



**Figura 1.** Aquisição da imagem 3D

Como estratégia, para obter um corte em um dos planos axial, coronal ou sagital, foram nomeados os vértices de um plano de visualização (00,01,10,11), Figura 2. Assim, os seguintes parâmetros foram utilizados na função para obter um corte desejado.

- 1 – Determinar qual o eixo do corte que seria executado (se x, y ou z).
- 2 – Determinar a posição em que seria o corte (de 0 até o tamanho na direção do corte).
- 3 – Determinar a ordem dos demais eixos, ou seja, qual a direção que iria percorrer primeiro.
- 4 – Determinar qual o vértice inicial para percorrer a imagem.



**Figura 2.** Plano de visualização. A) corte axial; B) corte sagital; C) corte coronal.

Com esta função, um profissional tem a liberdade de adequar-se à sua forma preferencial de visualização. No entanto, existem parâmetros pré-configurado para visualização axial (corte no eixo z), coronal (corte no eixo y) e sagital (corte no eixo x) do radiologista e do neurocirurgião.

## • Tarefa 2

Escreva uma função C para ajuste de brilho e contraste, usando largura e nível em **[0%, 100%]**. Avalie os resultados nas imagens de cortes axial, coronal e sagital da tarefa anterior.

Depois de obtido uma imagem 2D, é aplicado nela a função de ajuste de brilho e contraste para melhorar a visualização da mesma. Para isso, a implementação foi baseada no seguinte conteúdo teórico:

Sejam  $[l_1, l_2]$ ,  $l_1 \leq l_2$ , e  $[k_1, k_2]$  dois intervalos de cinza no conjunto de valores de  $I$  e  $J$ . A transformação linear (stretching linear) é definida por:

$$k = \begin{cases} k_1, & \text{se } l < l_1, \\ \frac{(k_2 - k_1)}{(l_2 - l_1)}(l - l_1) + k_1, & \text{se } l_1 \leq l < l_2, \\ k_2, & \text{se } l \geq l_2. \end{cases} \quad (1)$$

Alguns casos particulares são:

- Normalização em  $[0, H]$  (e.g.,  $H = 255$ ):  $k_2 = H$ ,  $k_1 = 0$ ,  $l_1 = l_{\min}$ , e  $l_2 = l_{\max}$ , onde  $l_{\min}$  e  $l_{\max}$  são os valores mínimo e máximo de  $\hat{I}$ .
- Negativo:  $k_2 = l_{\min}$ ,  $k_1 = l_{\max}$ ,  $l_1 = l_{\min}$ , e  $l_2 = l_{\max}$ .
- Largura & Nível (width & level):  $k_2 = H$ ,  $k_1 = 0$ , e  $l_1 < l_2$ , onde o nível  $\frac{l_1 + l_2}{2}$  altera o brilho e a largura  $l_2 - l_1$  altera o contraste.

As funções de normalização e negativo também foram implementadas, mas não serão discutidas neste relatório, somente seus resultados serão apresentados na próxima sessão.

O trecho abaixo mostra a principal parte da função de brilho **b** e contraste **c**, em que os níveis de ajustes são definidos pelo usuário.

```

I1 = b-(c/2);
I2 = b+(c/2);

for( y=0; y < I->ny; y++ ){
    for( x=0; x < I->nx; x++ ){
        aux = I->val[y][x];
        if ( aux < I1 )
            aux = 0;
        else if( aux >= I2 )
            aux = 255;
        else
            aux = (int) ( H/(I2-I1) * (aux-I1) );
        I->val[y][x] = aux;
    }
}

```

**Figura 3.** Parte do código que calcula o brilho e contraste de uma imagem.

• **Tarefa 3**

Escreva uma função  $C$  para composição de imagem colorida com sensação de transparência dos objetos de cortes axial, coronal e sagital da tarefa 1.

Caso seja feita uma segmentação na imagem médica, pode ser interessante sobrepor um corte da imagem original com um corte da segmentação para observar as áreas de cada objeto. Para a realização dessa tarefa utilizamos a imagem da libmc920 “brain.scn” e a imagem com três objetos segmentados “brain\_label.scn” também disponível na biblioteca.

A implementação dessa tarefa dá continuidade à tarefa 1, portanto é feito um corte na imagem médica e um corte na imagem segmentada, ambos com os mesmos parâmetros. Atribui-se uma cor (RGB) para cada objeto segmentado e as duas imagens podem ser mescladas, isso é feito transformando as cores RGB para YCgCo e substituindo a intensidade  $Y$  pelas intensidades do corte obtido da imagem médica.

Como os objetos da imagem segmentadas são identificados por números 1, 2 e 3, no caso da *brain\_label.scn*, tais números serão utilizados para a escolha da cor RGB de cada objeto. Visando um espaçamento aproximadamente uniforme, imutável e abrangendo todo o espectro possível do modelo RGB adotamos o seguinte método de geração de cores:

$$\begin{aligned} V &\leftarrow \text{Obj}/\text{Nobj} \\ V &\leftarrow (6 - 2)V + 1 \\ R(p) &\leftarrow H \max\{0, (3 - |V - 4| - |V - 5|)/2\} \\ G(p) &\leftarrow H \max\{0, (4 - |V - 2| - |V - 4|)/2\} \\ B(p) &\leftarrow H \max\{0, (3 - |V - 1| - |V - 2|)/2\} \end{aligned}$$

Ou seja, primeiramente, é atribuído um valor a  $V$  de acordo com o número do objeto  $\text{Obj}$  e o número total de objetos  $\text{Nobj}$ , no caso da *brain\_label.scn* temos um total de  $\text{Nobj} = 3$  objetos portanto os possíveis valores de  $V$  serão  $0.33$ ,  $0.66$  e  $1$ . Ao aplicar as outras operações o resultado é um conjunto de valores RGB com valor máximo de  $H$ , que escolhemos empiricamente como um valor próximo a intensidade média do corte da imagem médica para a junção conter os detalhes em escala de cinza e ser possível identificar com clareza as diferentes cores dos objetos.

Caso exista uma segmentação de 10 objetos os valores de  $V$ , na primeira linha, seriam  $0.1$ ,  $0.2$ ,  $0.3$ , ... e  $1$ , garantindo assim um espaçamento uniforme para facilitar a distinção entre as cores.

Com as cores no modelo  $RGB$  definidas, o próximo passo é transformar para o modelo  $YCgCo$  com o objetivo de separar a intensidade  $Y$  e substituí-la pela intensidade do corte cinza da imagem médica. Depois basta realizar a transformação de volta para o modelo  $RGB$  e salvar a imagem colorida. As transformações de ida e volta são descritas pelas equações:

$$\begin{bmatrix} Y(p) \\ Cg(p) \\ Co(p) \\ 1 \end{bmatrix} = \begin{bmatrix} 0.25 & 0.50 & 0.25 & 0.50 \\ -0.25 & 0.50 & -0.25 & \frac{H}{2} + 0.5 \\ 0.50 & 0.00 & -0.50 & \frac{H}{2} + 0.5 \\ 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix} \begin{bmatrix} R(p) \\ G(p) \\ B(p) \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} R(p) \\ G(p) \\ B(p) \\ 1 \end{bmatrix} = \begin{bmatrix} 1.00 & -1.00 & 1.00 & 0.00 \\ 1.00 & 1.00 & 0.00 & -\frac{H}{2} \\ 1.00 & -1.00 & -1.00 & H \\ 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix} \begin{bmatrix} Y(p) \\ Cg(p) \\ Co(p) \\ 1 \end{bmatrix}$$

Onde  $H$  é equivalente a  $2^b - 1$  onde  $b$  é a quantidade de bits da imagem.

- **Tarefa 4 (Extra)**

Escreva uma função  $C$  para desenhar no plano de visualização as arestas visíveis da cena (*wireframe*), assumindo faces opacas, para diferentes valores de  $\theta_x$  e  $\theta_y$ .

Para obter as arestas visíveis da cena, primeiro, é setado os oito vértices da cena. As transformações geométricas serão aplicadas a estes pontos, na seguinte ordem:

- 1 – Translação para o centro da cena.
- 2 – Rotação em X.
- 3 – Rotação em Y.
- 4 – Translação para o centro do plano de visualização.

As faces da cena são representadas pelos vetores  $(1, 0, 0, 0)$ ,  $(-1, 0, 0, 0)$ ,  $(0, 1, 0, 0)$ ,  $(0, -1, 0, 0)$ ,  $(0, 0, 1, 0)$ , e  $(0, 0, -1, 0)$ . Estas faces também sofrem transformações.

- 1 – Rotação em X.
- 2 – Rotação em Y.

Após a transformação das faces, seu produto interno com o vetor  $(0, 0, -1, 0)$  deve ser positivo para que a face seja visível. Em caso positivo, estas arestas são desenhadas no plano de visualização (usando a função de dda), obtendo uma imagem com o wireframe na rotação desejada.

- **Tarefa 5**

Escreva uma função em C para obter a imagem de um corte planar da cena 3D centrado em um ponto  $p_1 \in D_1$ , arbitrário e usando um vetor  $\vec{v}$  de visualização arbitrário.

Para a execução dessa tarefa foi criada uma função *GetSlice* que recebe o ponto central  $p_1$  e o vetor de visualização  $Vet$ , possuindo estruturas *Point* e *Vector* respectivamente, definidas na libmc920.

O método utilizado para extrair o corte foi criar um plano de visualização, aplicar transformações geométricas de translação e rotação para deixar este plano centralizado em  $p_1$  e com sua normal alinhada ao vetor de visualização e, por interpolação, obter as intensidades correspondentes na imagem médica. Sendo  $\phi$  a função de translação e rotação que alinha a cena no plano de visualização, tem-se que a função inversa  $\phi^{-1}$  é responsável por alinhar o centro de visualização na cena, de acordo com os parâmetros fornecidos pelo usuário.

A função geral  $\phi^{-1}$  que será aplicada em cada pixel do plano pode ser descrita como

$$\mathbf{T}(p_c)\mathbf{R}_x(-\theta_x)\mathbf{R}_y(-\theta_y)\mathbf{T}(-p'_c)$$

onde, da esquerda para a direita, o primeiro termo representa uma translação de tamanho  $p_c$ , o segundo uma rotação no eixo x, o terceiro outra rotação mas agora no eixo y e o último termo representa outra translação de uma quantidade  $-p'_c$ . Nessa atual tarefa consideraremos o primeiro termo uma translação da origem do sistema de coordenadas até o ponto  $p_1$  e o último termo uma translação do ponto central do plano de visualização até a origem, já os ângulos de rotação serão definidos de acordo com o vetor de visualização definido pelo usuário seguindo a regra da figura a baixo.

```

if (Vz>0) {
    ax = atan(Vy/Vz);
    ay = atan(Vx*cos(ax)/Vz);
}
else if (Vz<0) {
    ax = atan(Vy/Vz);
    if (ax!=0) ax = ax - PI;
    ay = atan((Vx*cos(ax))/Vz);
    if (ay!=0) ay = ay - PI;
}
else {
    if ((Vx==0) && (Vy!=0)) {
        ay = 0;
        if (Vy>0) ax = PI/2;
        else ax = -PI/2;
    }
    else if ((Vx!=0) && (Vy==0)) {
        ax = 0;
        if (Vx>0) ay = PI/2;
        else ay = -PI/2;
    }
    else { // Vx!=0 e Vy!=0
        if (Vy>0) az = -acos(Vx);
        else az = acos(Vx);
        ay = -PI/2;
    }
}
}

```

**Figura 4.** Parte do código que define os ângulos de rotação.

As matrizes de translação  $p_1$ , rotação no eixo x e rotação no eixo y ficam, respectivamente:

$$\mathbf{T}(p_1) = \begin{bmatrix} 1 & 0 & 0 & p_{1x} \\ 0 & 1 & 0 & p_{1y} \\ 0 & 0 & 1 & p_{1z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



A translação responsável por levar o plano de visualização para a origem do sistema de coordenada é semelhante a translação de  $p_1$  mas com os valores  $-D/2$  para as coordenadas  $x$  e  $y$  e zero para  $z$ , onde  $D$  é o tamanho da altura e largura do plano de visualização, correspondendo a diagonal da imagem médica.

Pipeline da função *GetSlice*:

- Criação das matrizes de translação;
- Definição dos ângulos de rotação;
- Criação das matrizes de rotação;
- Criação do plano de visualização;
- Aplicação de  $\phi^{-1}$  em cada ponto do plano de visualização:
  - Verificar se o ponto está no domínio da imagem médica;
  - Interpolação para encontrar a intensidade nesse ponto;
  - Salvar o valor de intensidade em uma imagem cinza;
- Retornar ou salvar a imagem cinza.

- **Tarefa 6**

Reformatação de um segmento de reta repetindo a tarefa anterior ao longo de pontos dessa reta, gerando um novo conjunto de dados tridimensional.

A função para a realização dessa tarefa recebe do usuário dois pontos e o número de cortes que devem ser retirados da imagem médica. A função da tarefa 5 será chamada em cada passo do segmento de reta entre os dois pontos fornecidos pelo usuário até totalizar a quantidade desejada. Cada corte é gravado no disco, o volume composto pelas imagens de saída não é isotrópico.

- **Tarefa 7**

Escreva uma função para calcular as projeções de máxima e média intensidade de uma cena de entrada, para um dado ponto de vista definido pelos ângulos  $0^\circ \leq \theta_x < 360^\circ$  e  $0^\circ \leq \theta_y < 360^\circ$  em graus.

Essa tarefa possui uma estrutura parecida com a implementada na tarefa 5, a principal diferença da estrutura está que o plano de visualização não será mais colocado dentro da cena, para cada ponto do plano de visualização será percorrida uma reta em direção a cena buscando um valor de intensidade máximo ou calculando a intensidade média ao longo dessa reta.

A matriz de translação do centro do plano de visualização será preenchida com  $-D/2$  nas três coordenadas, deixando-o bem afastado da cena. Utilizando a fórmula descrita a baixo encontraremos valores de  $\lambda$  de pontos que interceptam duas faces da cena ao longo de uma reta paralela a normal do plano de visualização na direção da cena. O algoritmo DDA percorrerá pontos do segmento de reta entre as faces, realizando uma interpolação a cada passo e salvando os valores máximos de intensidade ou a média do segmento.

Pipeline da função MIP:

- Criação das matrizes de translação;
- Definição dos ângulos de rotação;
- Criação das matrizes de rotação;
- Definição das faces da cena;
- Criação do plano de visualização;
- Aplicação de  $\phi^{-1}$  em cada ponto do plano de visualização:
  - Calcula o valor de  $\lambda$  para cada face da cena;
    - Encontrar dois  $\lambda$ s válidos que resultam em pontos dentro da cena;
  - Aplicar o algoritmo DDA para percorrer uma reta na direção da cena;
    - Interpolar cada ponto da reta e salvar o valor máximo ou médio da mesma;
- Retornar ou salvar uma imagem cinza.

### • Tarefa 8

Escreva uma função em C para gerar o *rendering* de textura das iso-superfícies internas a esses objetos, para uma dada direção de visualização.

Aqui foi implementado o modelo de iluminação de Phong, que simula como uma superfície refletiria luz utilizando combinações de reflexões difusas e especular, levando também em consideração a luz do ambiente e clareamento pela proximidade do objeto. O modelo simplificado pode ser aplicado utilizando a equação

$$L(p) = k_a L_a + D(p) (k_d \cos(\theta) + k_s \cos^{n_s}(2\theta))$$

onde  $L(p)$  é a intensidade total de um pixel,  $k_a$ ,  $k_d$  e  $k_s$  são constantes de balanceamento dos termos de iluminação do ambiente, reflexão difusa e especular, respectivamente, onde  $k_a + k_d + k_s = 1$  e  $D(p)$  é uma função inversamente proporcional a distância do pixel ao plano de visualização. O  $\theta$  é o ângulo entre a normal da superfície onde a luz está iluminando e o observador.

Boa parte do algoritmo do MIP (tarefa 7) foi utilizado, as principais mudanças ocorrem dentro da aplicação do DDA que, ao invés de salvar o valor máximo ou calcular a média, verifica se o pixel pertence à superfície de algum objeto, calcula a distância desse pixel até o plano de visualização e o valor aproximado do vetor normal da superfície nesse ponto.

Pipeline da função Rendering:

- Criação das matrizes de translação;
- Definição dos ângulos de rotação;
- Criação das matrizes de rotação;
- Definição das faces da cena;
- Criação do plano de visualização;
- Aplicação de  $\phi^{-1}$  em cada ponto do plano de visualização:
  - Calcula o valor de  $\lambda$  para cada face da cena;
    - Encontrar dois  $\lambda$ s válidos que resultam em pontos dentro da cena;
  - Aplicar o algoritmo DDA para percorrer uma reta na direção da cena;
    - Verifica os vizinhos do pixel procurando a superfície;

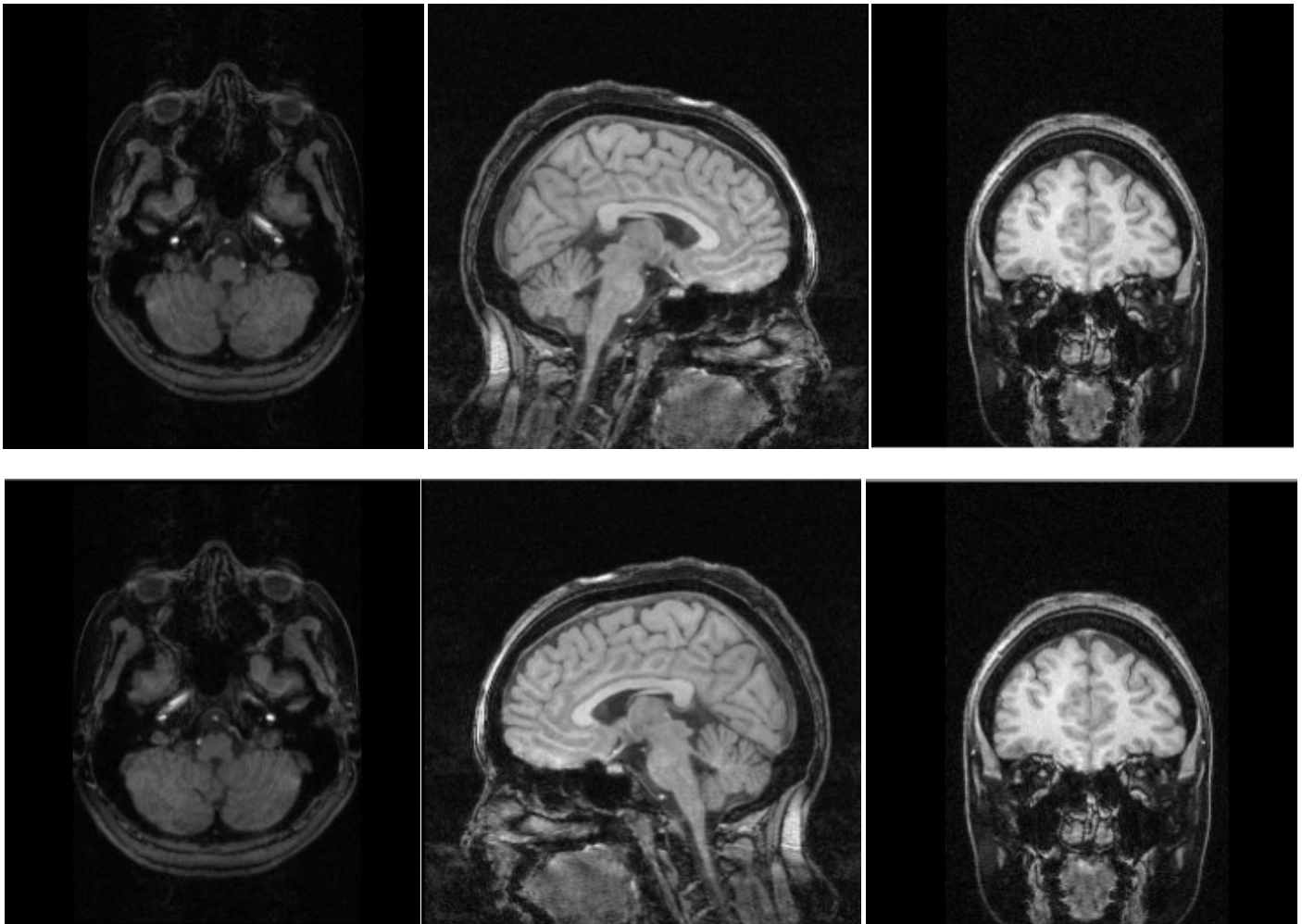
- Se achar um pixel válido da superfície:
  - Calcula a distância desse pixel até o plano de visualização;
  - Estima a normal e calcula o  $\theta$ ;
- Calcula o  $L(p)$  para todos os pixels;
- Retornar ou salvar uma imagem cinza.

A estimação da normal utilizada foi proposta pelo professor da disciplina e descrita pelo colega Azael Sousa, e é baseada no cálculo do gradiente em três dimensões do pixel que está sendo “visitado”. São apresentados resultados variando-se as diversas constantes da equação de Phong.

## Resultados

- Tarefa 1

Os resultados referem-se a visão do radiologista e neurologista.



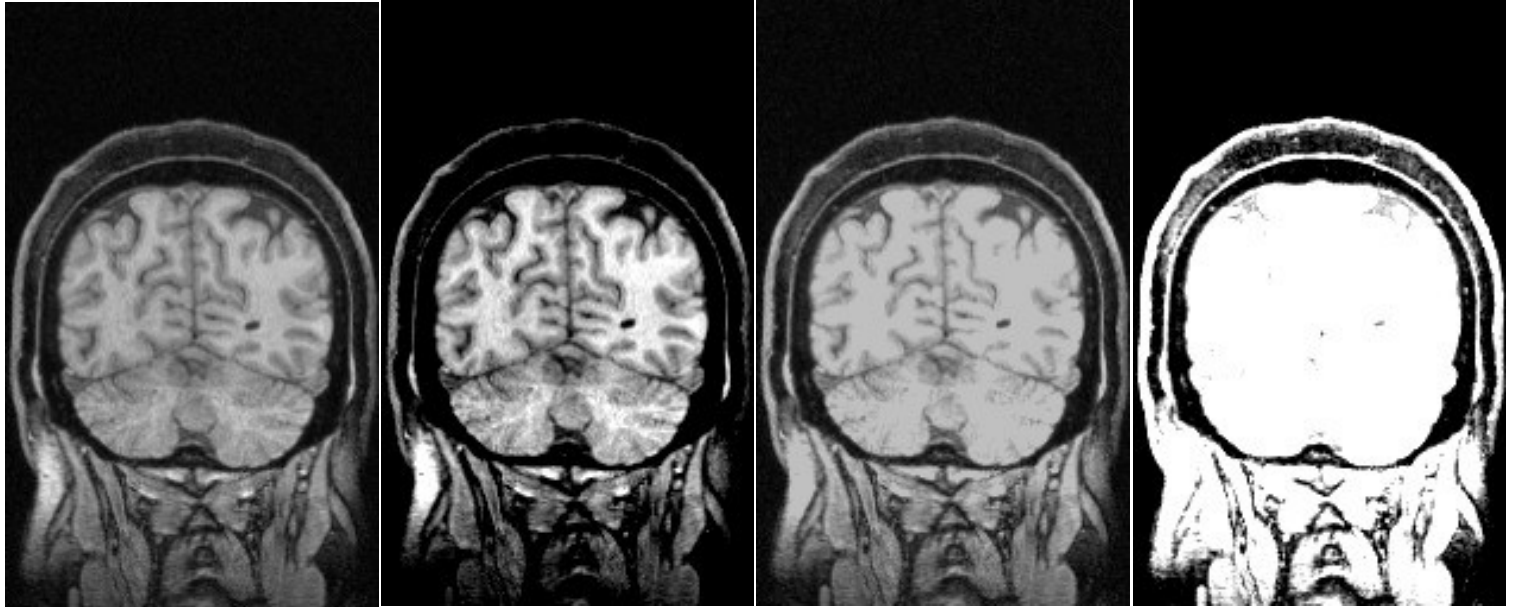
**Figura 5.** Imagem brain, corte na posição 75. A visão do radiologista em: A) axial; B) sagital; C) coronal. A visão do neurocirurgião em: D) axial; E) sagital; F) coronal.



**Figura 6.** Imagem skull, corte na posição 150. Visão do neurocirurgião: A) axial; B) sagital; C) coronal.

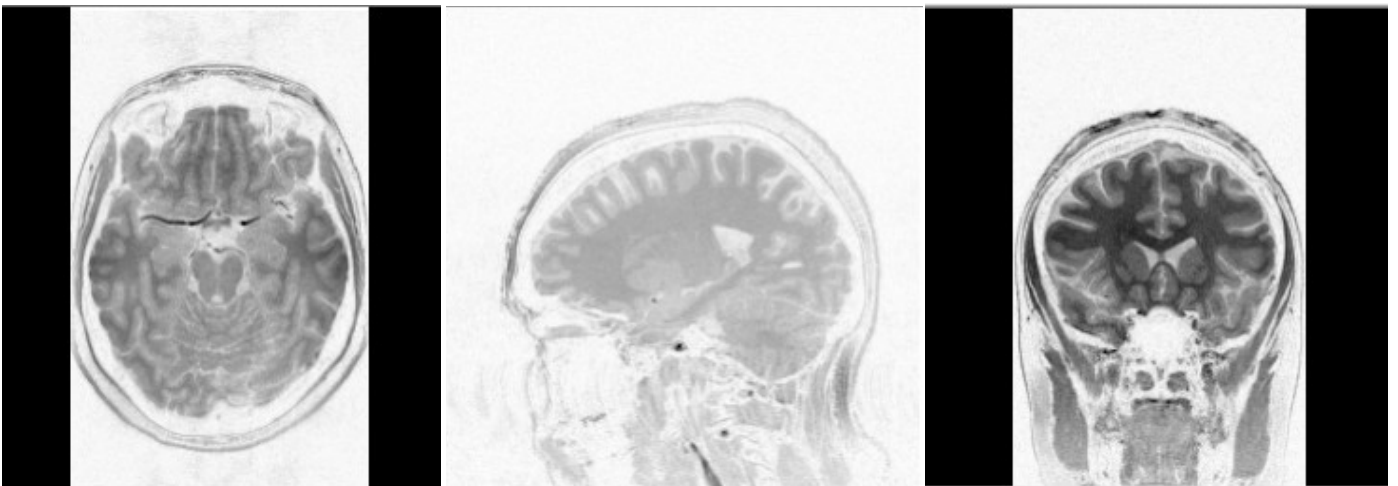
- **Tarefa 2**

O resultado obtido da função de brilho e contraste.



**Figura 7.** Brilho e Contraste da imagem brain, corte na posição 180. A) Imagem Original; B) Brilho: 50 Contraste: 50; C) Brilho: 80 Contraste: 20; D) Brilho: 80 Contraste: 80.

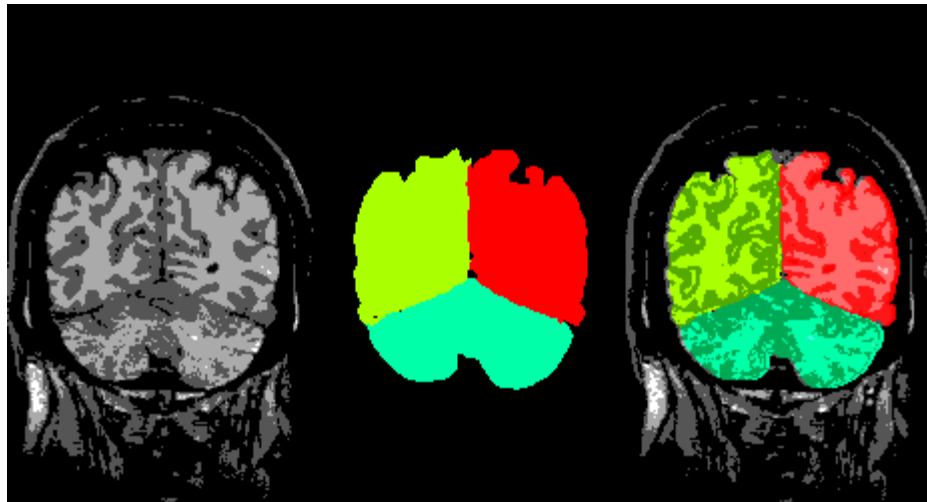
Também foi obtido o resultado da função do negativo da imagem do corte.



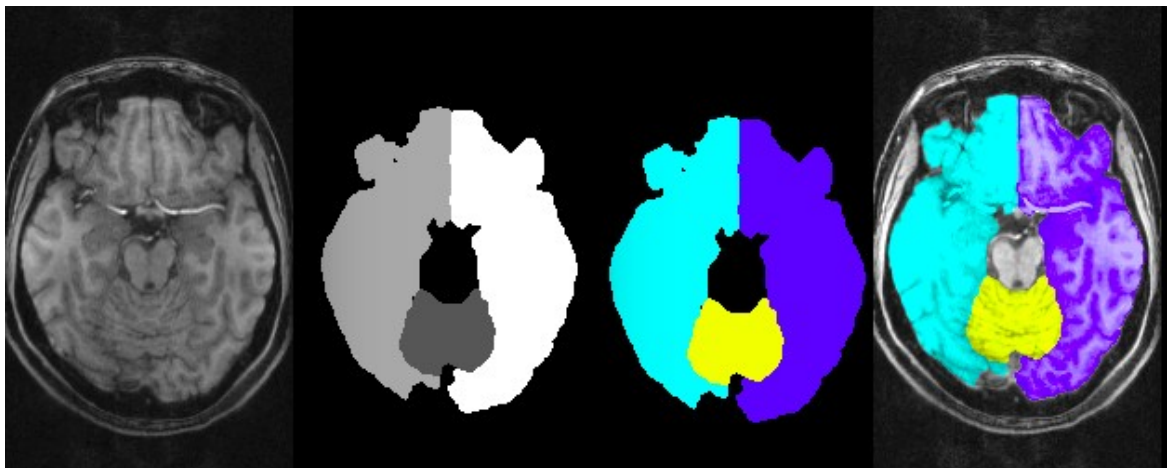
**Figura 8.** Negativo da imagem brain, corte na posição 100. Visão do neurocirurgião: A) axial; B) sagital; C) coronal.

- **Tarefa 3**

O resultado obtido da função de sobreposição do label de uma imagem no corte da imagem correspondente.



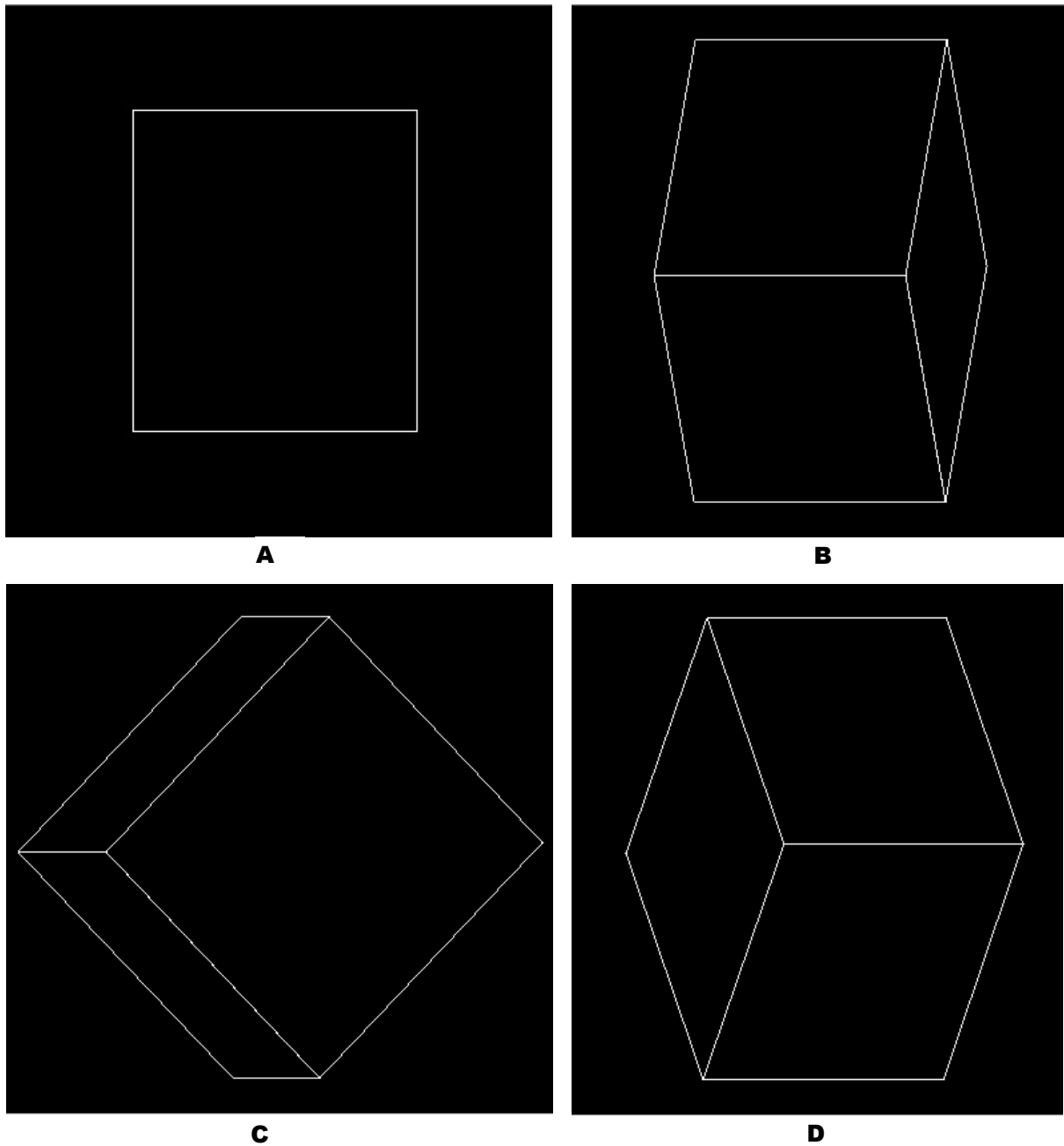
**Figura 9.** Esta imagem foi apresentada no início do semestre, em que a escala em cinza não estava boa e exibiu poucos tons de cinza, mas a aplicação das cores atingia o objetivo.



**Figura 10.** Resultado obtido do corte da imagem cinza corrigido e a sobreposição da label na imagem.

- **Tarefa 4 (Extra)**

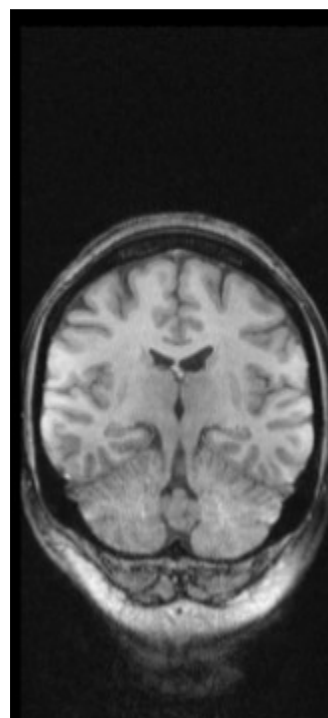
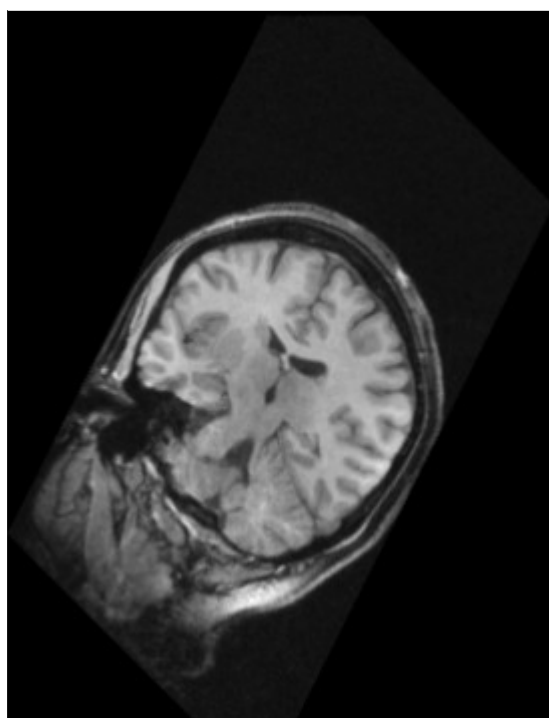
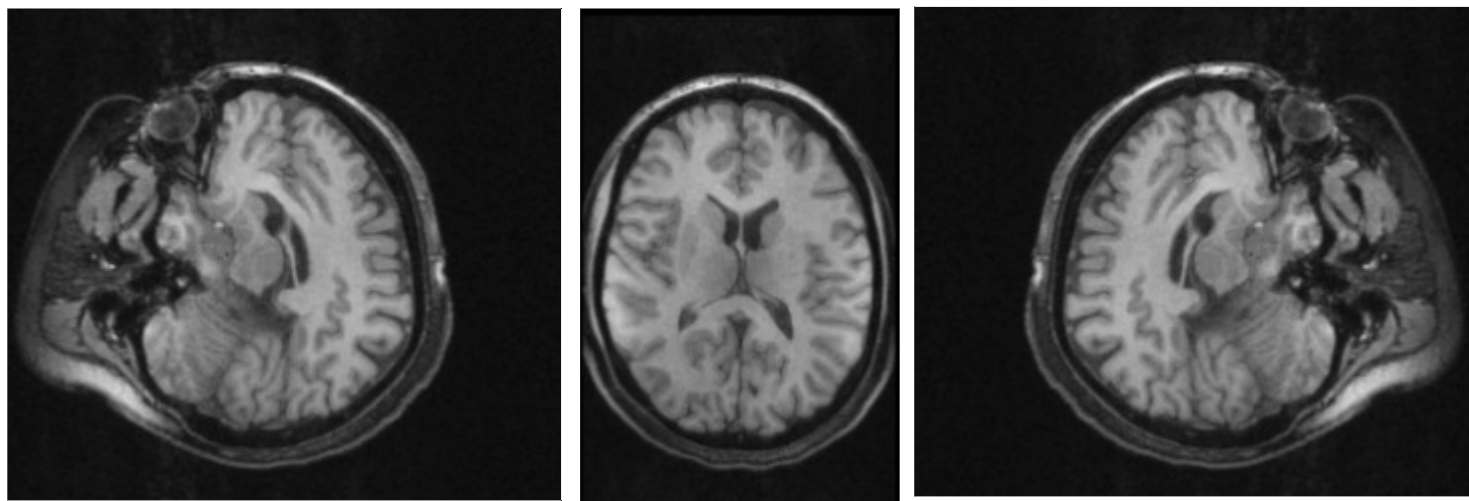
O resultado obtido refere-se as arestas visíveis depois de uma determinada rotação.



**Figura 11.** Arestas da imagem brain. A) Rx:0, Ry:0; B) Rx:45, Ry:10; C) Rx:45, Ry:70; D) Rx:45, Ry:160.

- **Tarefa 5**

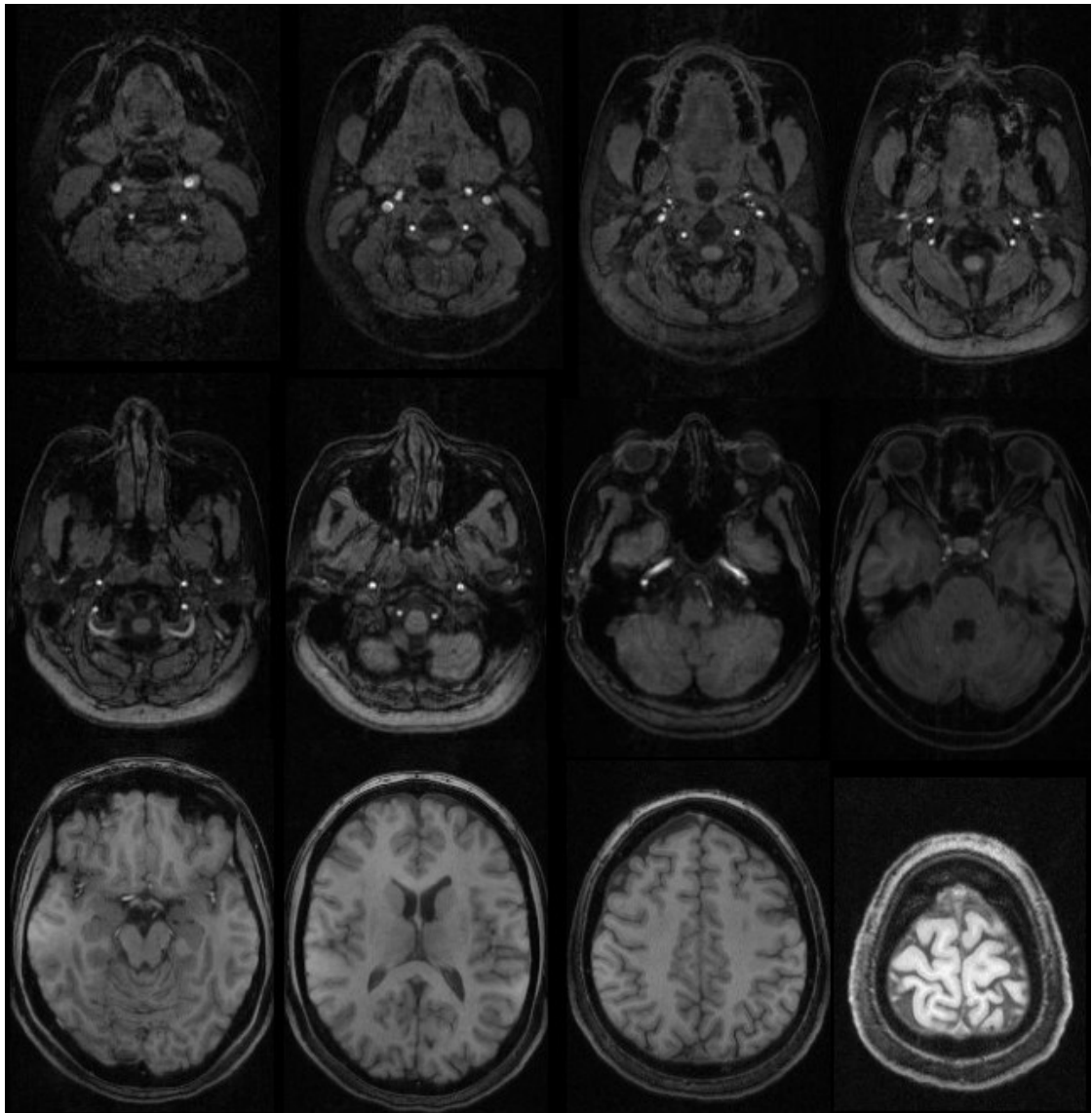
O resultado obtido do corte planar arbitrário  $p_1$  e visualização de acordo com um vetor.



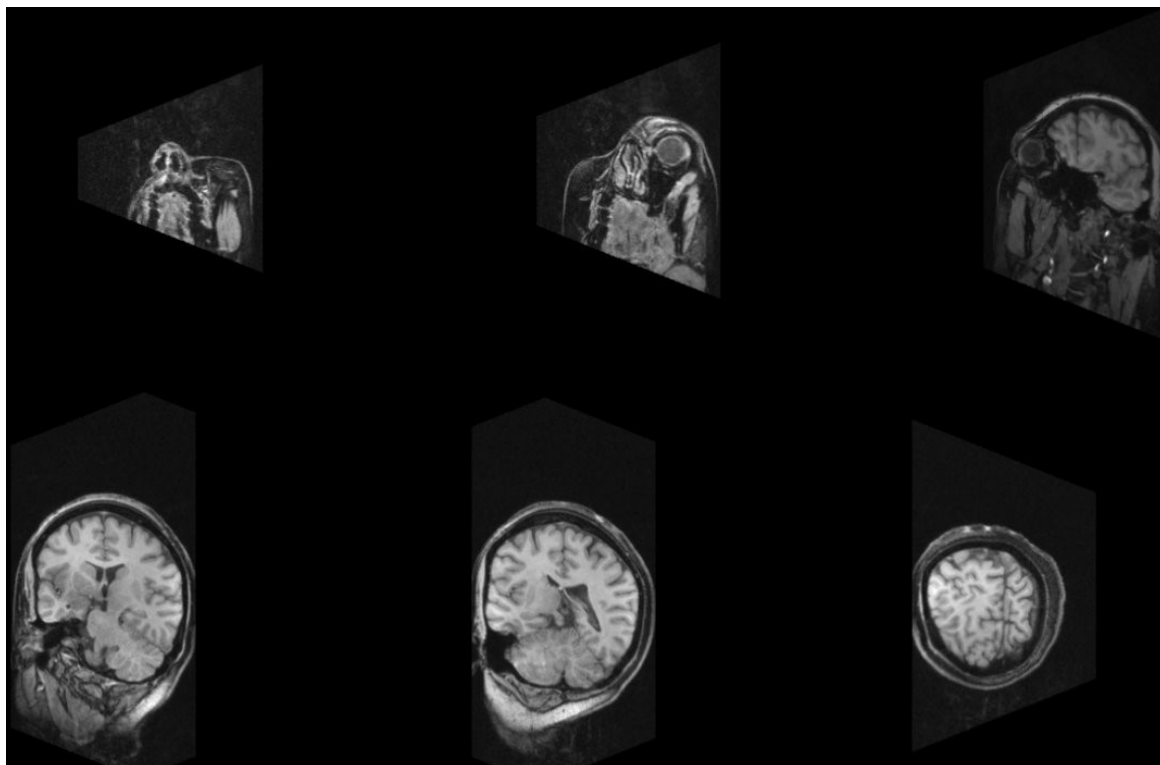


- **Tarefa 6**

O resultado obtido do volume gerado pela reformatação, um conjunto de  $N$  cortes entre dois pontos.



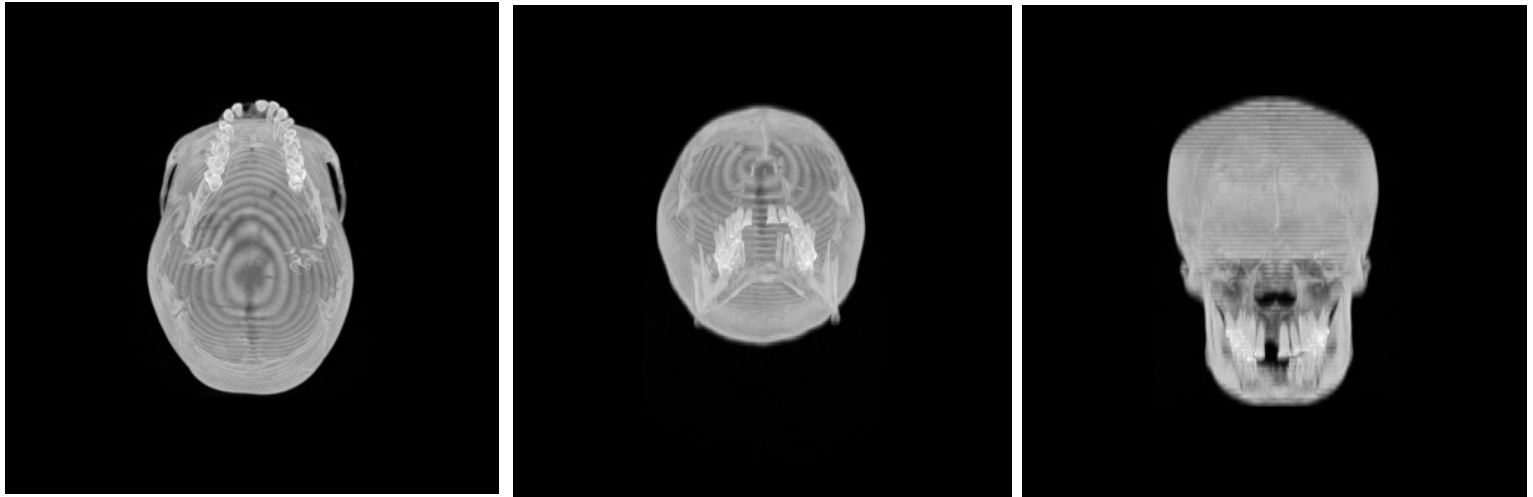
Volume gerado ao longo do eixo z da imagem médica.



Alguns dos cortes do volume gerado seguindo o segmento da diagonal principal.

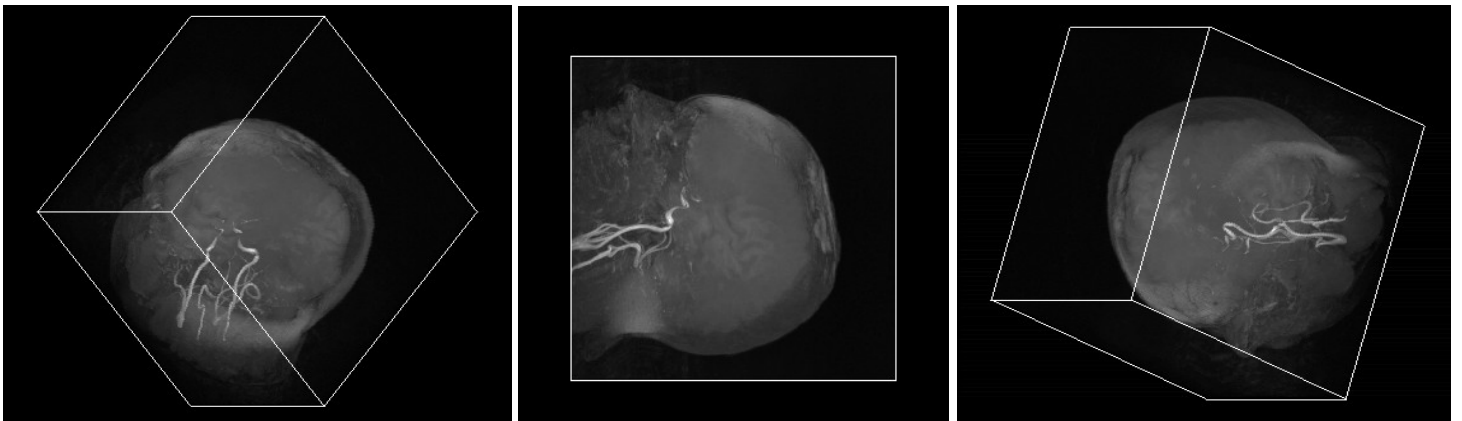
- **Tarefa 7**

Aqui é apresentado o resultado da projeção de intensidade máxima.



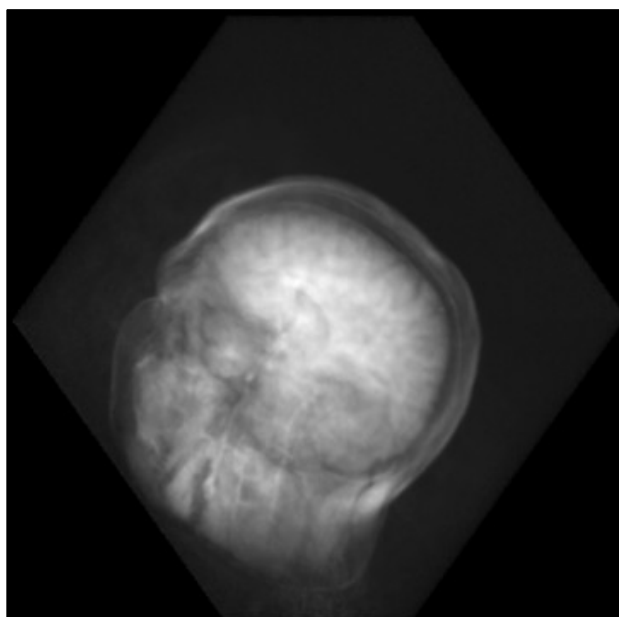
**Figura 3.** MIP da imagem skull. A) sem rotação; B) Rx:45, Ry:0; C) Rx:90, Ry:0.

O resultado desta tarefa também foi agrupado com a tarefa do wireframe.



**Figura 3.** MIP da imagem brain. A) Rx:0, Ry:90; B) Rx:45, Ry:45; C) Rx:160, Ry:45.

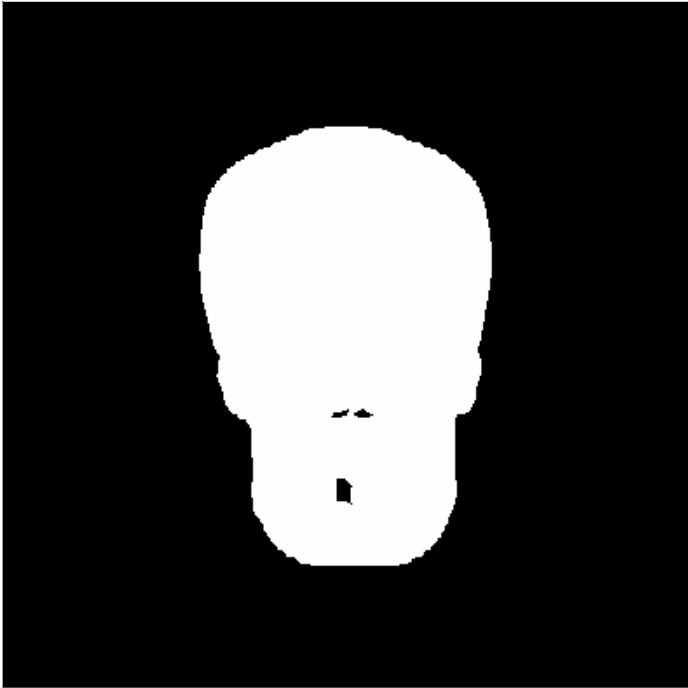
O resultado a seguir, refere-se a projeção de média intensidade.



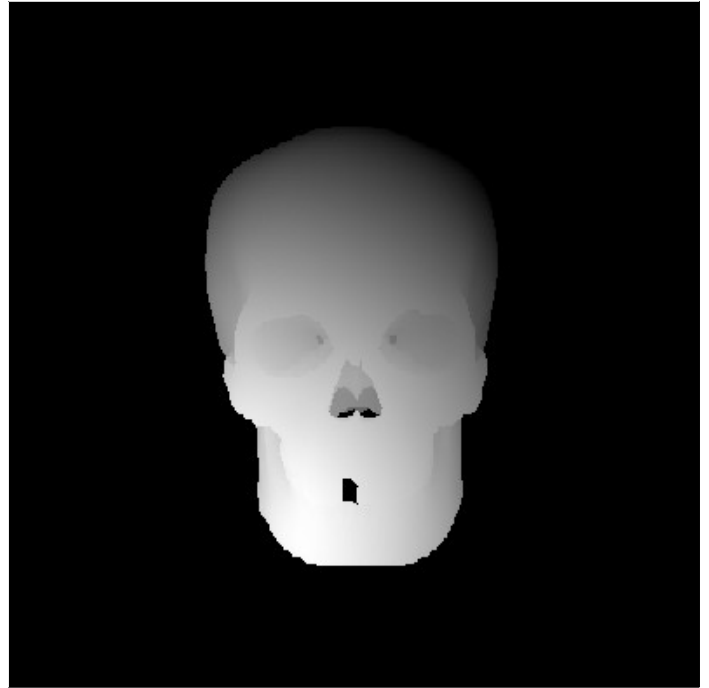
**Figura 3.** Projeção de média intensidade da imagem brain. Com Rx:45 e Ry:0.

- **Tarefa 8**

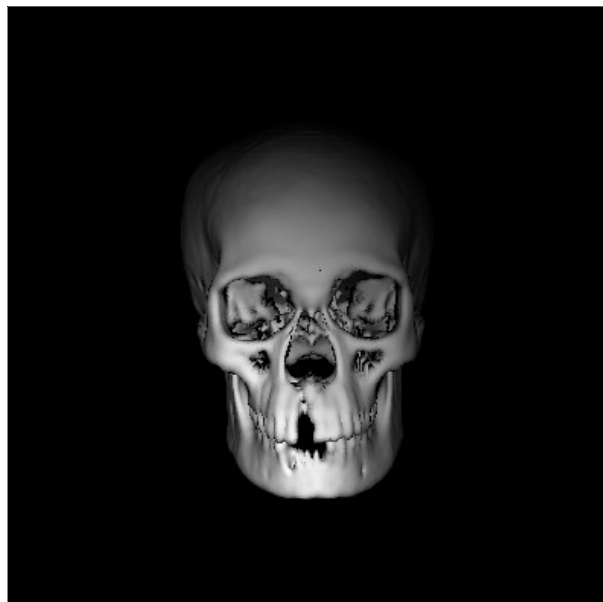
O resultado obtido do rendering de superfícies, aplicando o modelo de iluminação de Phong



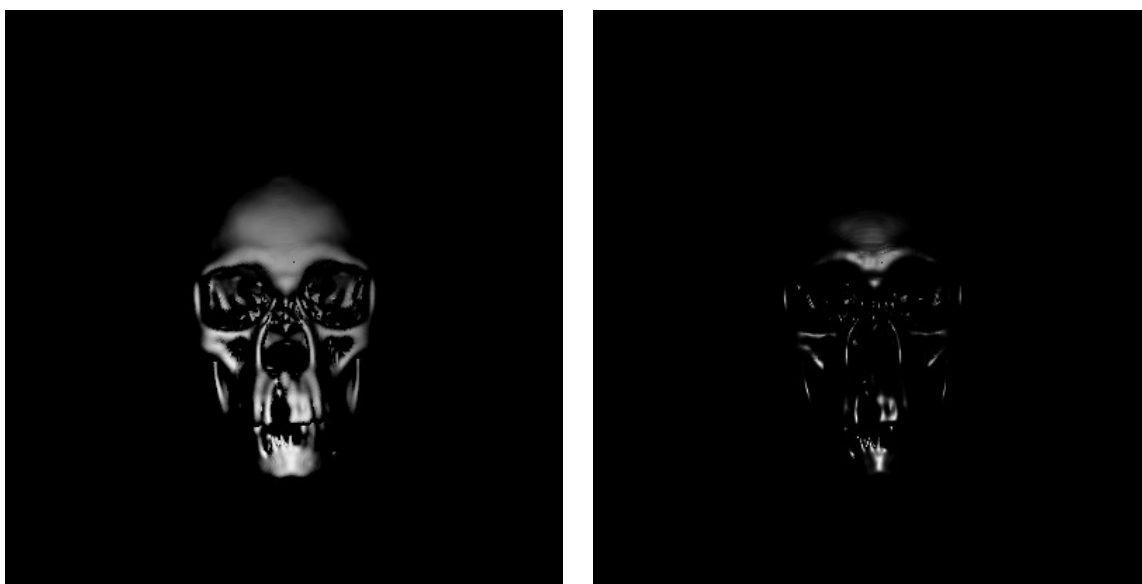
**Figura 7.** Imagem com apenas a luz do ambiente, primeiro termo, onde é exibido um valor igual de intensidade para todos os pixels da superfície encontrada.



**Figura 8.** Resultado da função de distância, quanto maior a distância menor a intensidade.



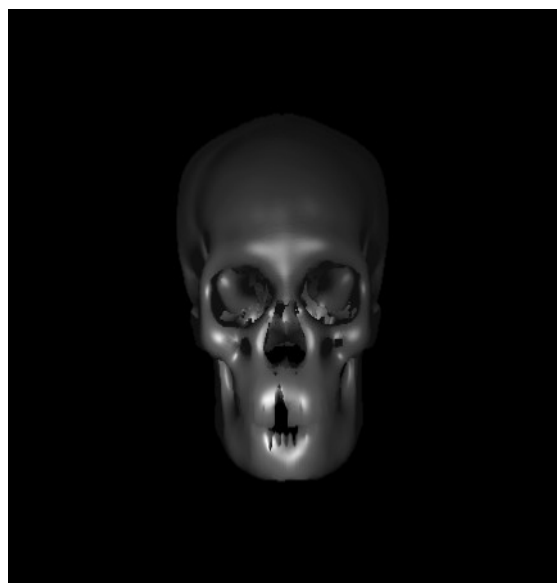
**Figura 9.** Resultado da reflexão difusiva.



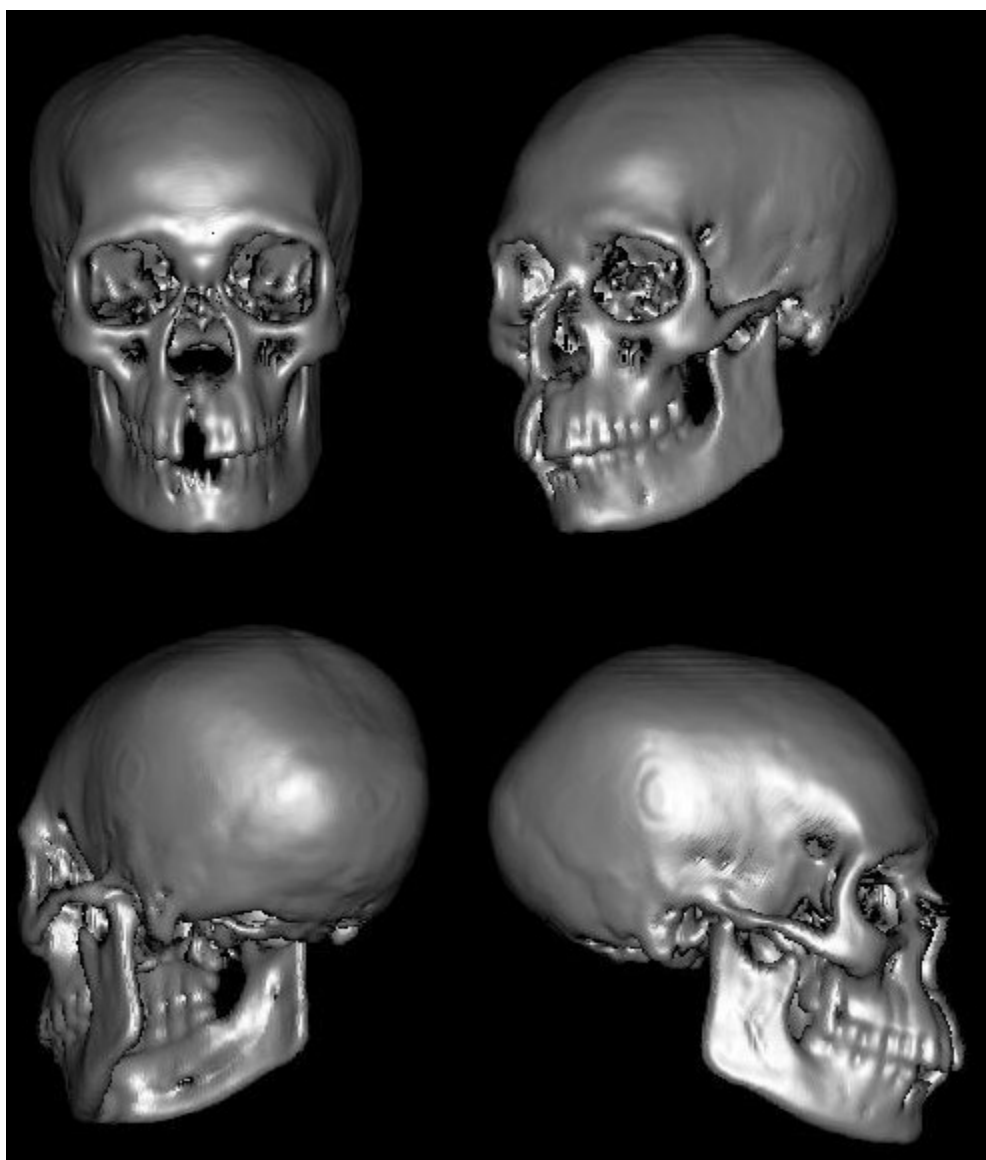
**Figura 10.** Resultados das reflexões especulares com valores de  $n_s$  igual a 1 (esquerda) e 10 (direita).



**Figura 11.** Resultado com raio = 5



**Figura 12.** Resultado com raio = 20



## Conclusão

- **Aprendizado**

O desenvolvimento do projeto proporcionou a aprendizagem de técnicas de manipulação de imagens, especificamente algumas das técnicas necessárias no tratamento de imagens médicas para que o profissional de saúde consiga interpretá-las da forma que está mais habituado, evitando equívocos.

Os métodos desenvolvidos podem ser abstraídos e reaproveitados para a resolução de problemas de outras áreas e outros tipos de imagem, já que a lógica principal de manipulação de imagens permanece a mesma em diferentes casos.

As principais áreas abordadas foram imagens cinza, imagens volumétricas (3D), imagens coloridas e transformações radiométricas, entre espaços de cores e geométricas. Além de outros conceitos matemáticos e de álgebra linear necessários para a implementação das tarefas que foram aperfeiçoados como interpolação e operações matriciais.

- **Dificuldades**

Cada integrante da dupla apresentou dificuldades diferentes, algumas delas são: dificuldades em entender o problema de cada tarefa, visto que cada tarefa tratava de um conceito novo e com um grau de complexidade mais elevado; dificuldades no início do semestre na utilização da biblioteca fornecida pelo professor, e na manipulação das imagens médicas de 8 e 16 bits, que apresentavam resultados contraditórios. Como houve uma grande abertura à diálogos e resolução de dúvidas, com o professor e colegas de classe, essas e outras dificuldades puderam ser superadas.

Infelizmente, devido à algumas dessas dificuldades e necessidade de dividir o tempo de estudo com outras matérias, a tarefa 6 e a tarefa 8 ficaram incompletas. A primeira por não garantir um volume isotrópico e a segunda por não oferecer a possibilidade de exibição de objetos transparentes.



## Referências

Material da disciplina Visualização de Imagem Volumétrica (MO815B/MC871A) do segundo semestre de 2016. Disponível em: <http://ic.unicamp.br/~afalcao/mo815-3dvis/index.html>.

**Digital image processing for medical applications**, Geoff Dougherty. New York, NY : Cambridge University Press, c2009.