



# Computação Gráfica - Trabalho Prático II

Prof. Márcio Sarroglia Pinho / 2014-I

Augusto Weiand - 14190491-2



## DESCRIÇÃO

O objetivo deste trabalho é realizar a segmentação de estruturas de um dente em imagens de MicroCT, usando técnicas de segmentação por limiar e técnicas de segmentação por textura. Para tanto, você deverá desenvolver um programa que lê a imagem e identifica as estruturas (dentina e canal) colorindo-as com cores distintas.

A cor escolhida para uma estrutura deve ser aplicada a todas as partes desta estrutura na imagem. Além disso, o programa deverá informar a quantidade de pixels de cada estrutura.

- ◆ **Parte 0 (02/06/14) – Entrega do Ground Truth**

Cada aluno deve realizar a segmentação manual de 5 imagens, colorindo, manualmente, as estruturas da seguinte forma:

- Dentina: verde; Canal: vermelho; Pinos: azul; Fundo: preto.

As imagens resultantes devem ser disponibilizadas ao professor por meio de um link, que será distribuído aos demais colegas da turma. Estas imagens servirão como ground truth, para os algoritmos de segmentação a serem implementados.

Para obter os dados, acesse este link: <http://www.inf.pucrs.br/~pinho/CG-PPGCC/T2-2014-1/Dados/>

- **Parte 1 (23/06/14) - Entrega da Segmentação por Limiar:**

Esta parte consiste na segmentação das estruturas dos dentes usando algoritmos de segmentação por limiar. Antes de aplicar estes algoritmos, o programa deve aplicar filtros na imagem a fim de pertimir uma melhor segmentação.

O programa deverá usar o ground truth para avaliar o resultados de seus algoritmos.

- **Parte 2 (07/07/14) - Entrega/Apresentação da Segmentação por Texturas:**

Esta parte consiste na segmentação das estruturas dos dentes usando algoritmos de segmentação por análise de texturas. Para saber mais sobre análise de texturas, consulte [este material](#).

O programa deverá usar o ground truth para avaliar o resultados de seus algoritmos.

### Carga de Imagens:

Os algoritmos de tratamento das imagens deverão ser implementados, sem o uso de bibliotecas como OpenCV, JavaCV, ImageJ, ou outras com capacidades semelhantes.

Para a carga das imagens, pode ser utilizada a biblioteca [ImageClass](#), a partir do programa disponível neste [link](#), ou outro que tenha capacidades semelhantes.

Link para o Repositório: <https://github.com/aweiand/SegmentaDente/tree/stable>

Imagens tratadas: <https://github.com/aweiand/SegmentaDente/tree/stable/Parte1/dataOUT>

Relatório: <https://github.com/aweiand/SegmentaDente/tree/stable/Parte1/dataREPORT>

## RESULTADOS

- Parte 1:

Para o desenvolvimento do sistema foi utilizado como aporte as bibliotecas ImageClass e SOIL, para a carga de imagens. Para o tratamento com filtros e a segmentação, foram desenvolvidos algoritmos próprios, baseando-se nos slides de aula do prof. Márcio.

Com a utilização da biblioteca ImageClass foi necessário converter as imagens originais para outro formato, bem como outra paleta de cores, pois as mesmas se encontravam originalmente em formato *bitmap* em tons de cinza, inviabilizando assim sua utilização e posterior salvamento em formato *rgb*, desta forma elas foram convertidas para *Portable Network Graphics - PNG*, com paleta *rgb*, possibilitando assim abri-las com a biblioteca, manipula-las de forma adequada e assim salvar o resultado da segmentação em uma nova imagem de formato *rgb*.

Foram implementados diversos algoritmos de segmentação por linear nas imagens a fim de que fosse possível a segmentação das partes de uma melhor forma. Desta maneira os algoritmos que foram utilizados foram os seguintes:

*Definição de intervalo de tons de cinza:* De modo de que a imagem se torne mais simples de segmentar, ela foi tratada para que tenha somente 4 tons de cinza e branco, os intervalos utilizados foram os seguintes: menores que 50, foram convertidos em 25; valores entre 50 e 100, em 75; valores entre 100 e 200, em 125; finalmente valores maiores que 200, em 255.

*Média dos vizinhos:* O algoritmo de média dos vizinhos foi aplicado para que fosse possível identificar os pinos da imagem. Ele foi aplicado com um kernel de tamanho 5, somando-se a intensidade dos pontos do kernel e após criando a sua média para a substituição na imagem de destino pela média obtida. Assim, caso a média resultante seja maior que 100, o ponto torna-se preto, caso contrário torna-se branco. Após isso aplica-se uma função que pinta os pontos de azul na imagem original de acordo com estes tratamentos na imagem de destino.

*Expansão de Pinos:* Como as imagens possuem algumas falhas e os algoritmos de filtro não foram capazes de corrigi-las, foi implementado um algoritmo de expansão, que utiliza uma matriz de expansão de 10x10, com o ponto central no topo e centralizado no meio da matriz. Expandindo os pontos com cor azul, conforme encontrados na imagem original.

*Limpeza de impurezas do fundo:* Este algoritmo utiliza a média dos vizinhos do ponto atual para criar um filtro de intervalo, caso o intervalo esteja entre 10 e 45 e não seja azul, este ponto será pintado de preto, caso contrário sera mantida a sua cor original. Com isto conseguiu-se limpar diversas impurezas do fundo preto.

*Encontrando a dentina:* Com a imagem tratada previamente e já encontrados os pinos, percorre-se a imagem procurando intervalos da média dos vizinhos que sejam maiores que 10 e não sejam azuis, estes são pintados de verde, os demais são mantidos de acordo com suas cores originais.

*Encontrando os canais:* A imagem é percorrida na horizontal até que se encontre o primeiro pixel preto após um verde, com isso um contador é inicializado, verificando se o próximo pixel continua sendo preto, até que se encontre um pixel verde em sua sequência.

Após este laço, é verificado se o contador está dentro do intervalo estabelecido, pelo qual foi definido os valores mínimos e máximos que um canal pode ter, assim, se dentro dos parâmetros este trecho é pintado de vermelho.

### RELATÓRIO DO SISTEMA

O sistema gera um relatório (<http://goo.gl/XMn3QU>) com o número de pixels e o percentual de cores *RGB* encontradas na imagem tratada por ele e na imagem de mesmo nome no Ground Truth. De acordo com este relatório, foi possível verificar que o algoritmo tem um percentual de acerto que difere em média 1% apenas no que diz respeito aos tons azuis e verdes. Com relação aos pontos vermelhos, os mesmos tem um percentual de similariedade com o Ground Truth mais baixo em virtude de que não foi possível encontrar uma solução ideal para que sejam encontrados de forma eficiente estes ponto da imagem (canal), para então pinta-los.

Desta maneira o sistema consegue segmentar as imagens em pinos, dentina e canal de forma que fique claro para um especialista realizar uma análise aceitável, porém não perfeita, pois a imagem resultante, apesar dos filtros aplicados, não ficou perfeitamente segmentada, resultando em alguns pontos verdes e vermelhos fora de seu escopo natural, juntamente com os pinos.

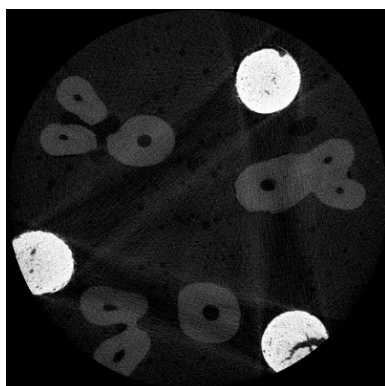


Imagem Original

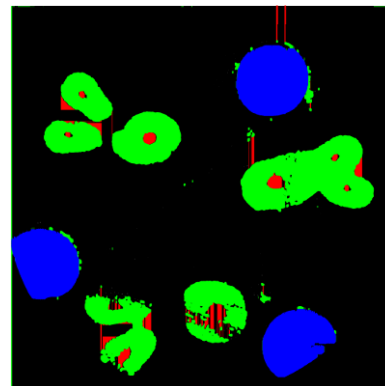


Imagem Tratada

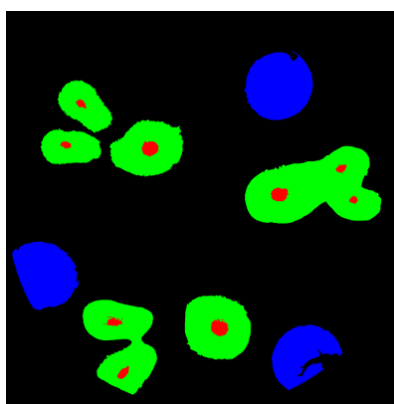


Imagem Ground Truth