

# Universidade Federal de Ouro Preto – DECOM/ICEB

BCC327 – Computação Gráfica

Trabalho Prático (TP1 e TP2)

Tema: Visualização Interativa e Realista de Árvores Arteriais  
Geradas pelo Método CCO

## 1. Contexto e Motivação

O método **CCO (Constrained Constructive Optimization)** é utilizado para gerar modelos de árvores arteriais em 2D e 3D a partir de princípios fisiológicos e geométricos. Esses modelos podem ser representados por arquivos contendo informações geométricas (posição dos vértices), topológicas (conectividade entre ramos) e morfométricas (raio de cada segmento). **Os dados de saída serão fornecidos em formato VTK (Visualization Toolkit).** O objetivo deste trabalho prático é desenvolver um sistema de **visualização interativa e realista** desses modelos, utilizando qualquer biblioteca ou API gráfica (*OpenGL, WebGL, Unity, Blender, Three.js*, entre outras). O trabalho será dividido em duas etapas: **TP1 (modelos 2D)** e **TP2 (modelos 3D)**.

As Figuras 1 e 2 ilustram exemplos de modelos 2D com diferentes números de segmentos terminais, enquanto as Figuras 3 e 4 apresentam exemplos de modelos 3D com distintos segmentos terminais

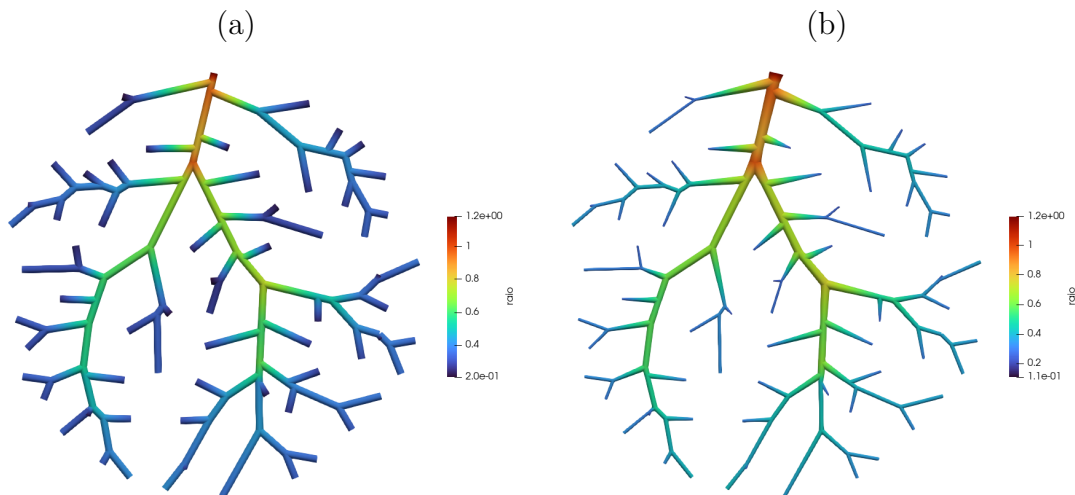


Figura 1: Exemplo de modelo 2D de árvore arterial com 64 segmentos terminais. (a) segmento cilíndrico (tubo) com raio fixo, (b) segmento cilíndrico (tubo) com raio variável.

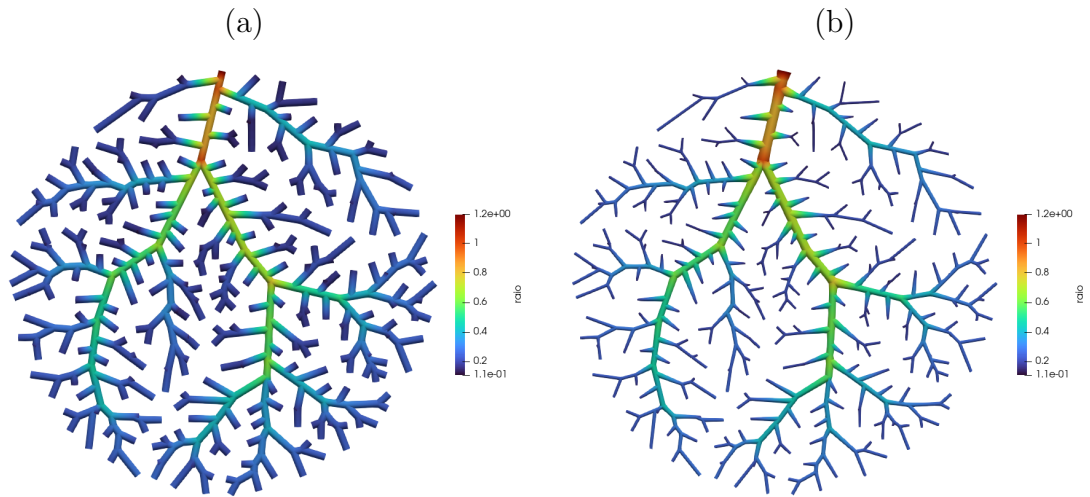


Figura 2: Exemplo de modelo 2D de árvore arterial com 256 segmentos terminais. (a) segmento cilíndrico (tubo) com raio fixo, (b) segmento cilíndrico (tubo) com raio variável.

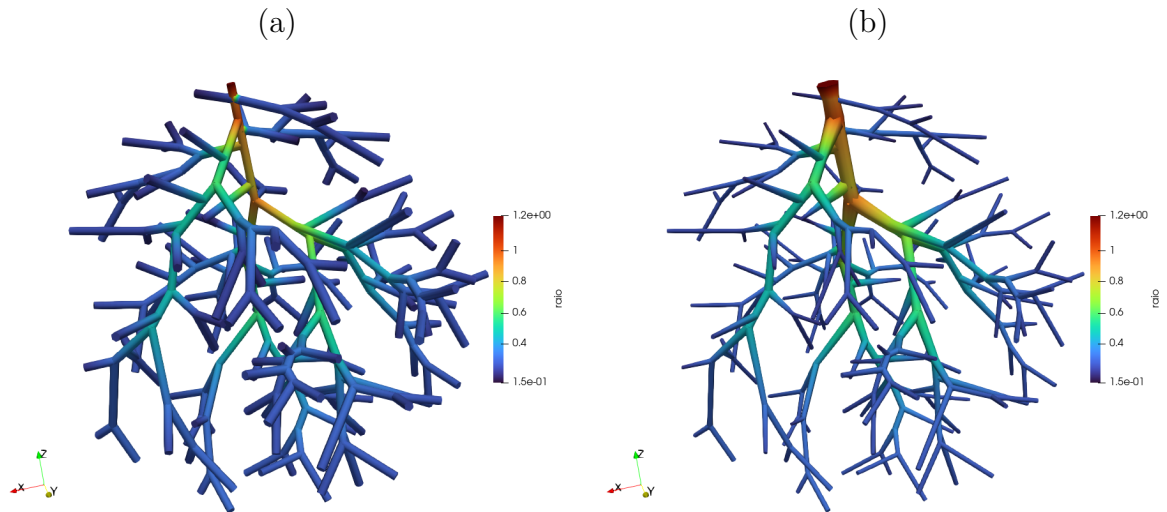


Figura 3: Exemplo de modelo 3D de árvore arterial com 128 segmentos terminais. (a) segmento cilíndrico (tubo) com raio fixo, (b) segmento cilíndrico (tubo) com raio variável.

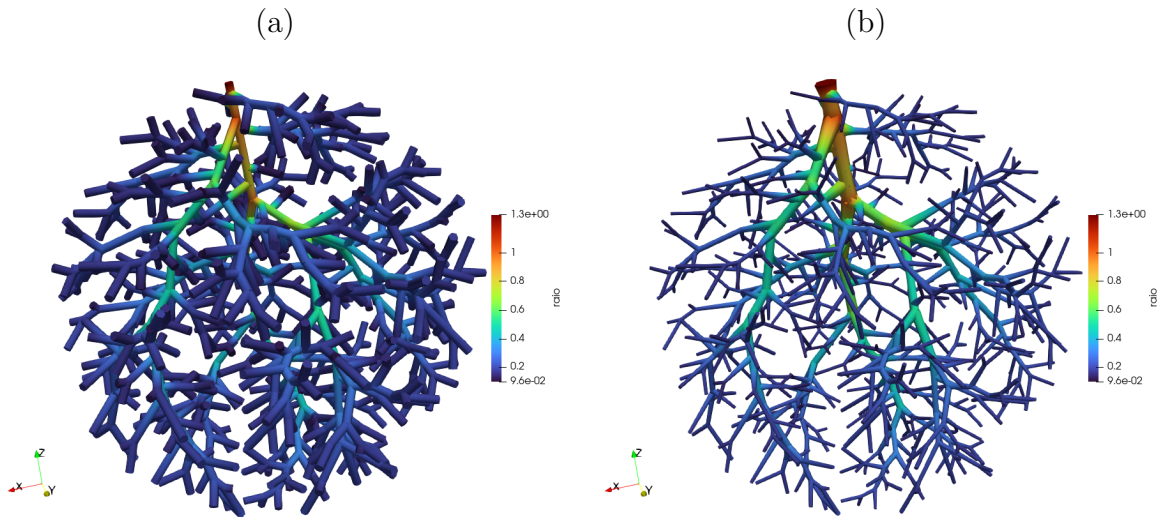


Figura 4: Exemplo de modelo 3D de árvore arterial com 512 segmentos terminais. (a) segmento cilíndrico (tubo) com raio fixo, (b) segmento cilíndrico (tubo) com raio variável.

## 2. Estrutura do Trabalho

### TP1 – Visualização e Transformações 2D

Entrega: 15/12/2025

#### Objetivos:

- Ler e representar dados geométricos 2D da árvore arterial (coordenadas e raios) a partir de arquivos **VTK**;
- Exibir a árvore 2D em um ambiente gráfico interativo;
- Aplicar transformações geométricas interativas (translação, rotação, escala) sobre o modelo;
- Implementar visualizações usando **projeções ortográficas**;
- Permitir visualização incremental do crescimento da árvore (utilizando os arquivos parciais gerados);
- (Opcional - Bônus) Implementar um algoritmo de recorte de segmentos de reta (*Cohen–Sutherland*, *Liang–Barsky* ou *Cyrus–Beck*).

#### Entregáveis:

1. Código-fonte comentado e executável;
2. Relatório técnico (2–4 páginas) com:
  - descrição do formato dos dados e estrutura de leitura;
  - explicação da escolha da API gráfica e de como as **transformações 2D foram implementadas matematicamente**;
  - capturas de tela das diferentes visualizações (incluindo o recorte de segmentos);
3. Vídeo curto (30–60 s) demonstrando a execução e a interatividade.

## TP2 – Visualização e Renderização 3D

Entrega: 23/02/2026

### Objetivos:

- Ler e representar dados 3D (coordenadas, conectividade e raios) a partir do arquivo **VTK** final;
- Modelar e exibir os ramos como **cilindros** ou segmentos de espessura variável, utilizando primitivas geométricas 3D;
- Implementar **projeção perspectiva** e controle de **câmera orbitante** (controle do observador);
- Aplicar **pelo menos dois** modelos de iluminação diferentes (*Flat*, *Gouraud* ou *Phong*);
- Implementar **transparência** e **remoção de superfícies escondidas (Z-buffer)**;
- Permitir **interação 3D**, como a seleção de um segmento de vaso para exibir suas propriedades (raio, comprimento, etc.);
- Criar animação temporal mostrando o crescimento da árvore arterial em 3D.
- (Opcional - Bônus) Inserir **texturas** ou aplicar efeitos de renderização avançados (*Ray Casting* / *Ray Tracing*).

### Entregáveis:

1. Código-fonte funcional e documentado;
2. Relatório técnico (4–6 páginas) descrevendo:
  - pipeline de renderização 3D implementado;
  - modelos de iluminação, **como a normal de cada superfície foi calculada** e justificativas da escolha;
  - comparações visuais e de desempenho entre os modos de iluminação;
3. Vídeo de demonstração (1–2 minutos) mostrando a iluminação, a câmera e a animação do crescimento.

## 3. Ferramentas sugeridas

O aluno pode escolher a tecnologia de sua preferência:

- **C++/OpenGL**: usando bibliotecas GLUT, GLFW, GLEW e GLM;
- **WebGL** ou **Three.js**: execução direta em navegador;
- **Unity (C#)** ou **Blender (Python API)**;
- Outras APIs serão aceitas mediante aprovação prévia.

## 4. Critérios de Avaliação

Critério	TP1 (2D)	TP2 (3D)
Leitura e estruturação dos dados (incluindo VTK)	2,0	–
Transformações e projeções (2D)	3,0	–
Iluminação, Transparência e Z-Buffer	–	3,0
Animção / Interação (2D/3D)	2,0	1,0
Qualidade visual e fidelidade geométrica (Modelagem 3D)	1,0	2,0
Relatório e vídeo	2,0	2,0
<b>Total</b>	<b>10,0</b>	<b>10,0</b>

## 5. Observações gerais

- Os trabalhos podem ser realizados individualmente ou em grupos de até 5 alunos;
- Códigos copiados entre grupos implicarão nota zero. **Todas as partes implementadas devem ser originais do grupo;**
- Códigos copiados da internet devem ser devidamente referenciados e creditados no relatório; caso contrário, a atividade receberá nota zero;
- Todas as bibliotecas externas utilizadas devem ser referenciadas e creditadas no relatório;
- O relatório deve ser entregue em formato PDF com autoria identificada;
- A entrega será feita via ambiente virtual da disciplina: Moodle.

**Prof. Rafael Alves Bonfim de Queiroz**  
Departamento de Computação – ICEB/UFOP