Projeto Final de Programação

Gerador Automático de Funções de Transferência

Orientador: Waldemar Celes

Aluno: Rustam Mesquita

Matrícula: 1512345

Julho / 2016

1. Introdução

Em visualização volumétrica, função de transferência é o nome dado ao mapeamento feito entre dados do volume e atributos ópticos, como cor e opacidade. Esse mapeamento permite destacar regiões de interesse, gerando uma visualização mais clara do volume. No entanto, a obtenção manual de uma boa função de transferência exige o conhecimento de como os dados variam no volume e paciência, já que esse método é consistido basicamente de tentativa e erro.

Tendo em vista a necessidade de gerar funções de transferência para a dissertação do mestrado e a dificuldade natural de realizar esta tarefa manualmente, optou-se por implementar um método de geração automática de funções de transferência como projeto final de programação. Como a renderização de volumes é por si só uma outra área de estudo, o projeto será integrado a um visualizador volumétrico a parte, de código aberto. Desta forma, não só o projeto está altamente alinhado à dissertação, como também será útil para os usuários do visualizador ao qual será integrado.

O projeto fará uso do método de geração automática descrito no artigo “Semi-automatic Generation of Transfer Functions for Direct Volume Rendering”, Kindlmann e Durkin, 1998. Portanto, os resultados serão validados pelo uso de volumes utilizados no artigo citado.

1. Requisitos Funcionais

* O projeto deve ser integrado a um visualizador volumétrico.
* O programa deve gerar como saída uma função de transferência, tendo como entrada um volume de dados.
* O programa deve passar ao visualizador a função de transferência gerada.
* O programa não deve interferir no uso do visualizador por parte do usuário, de forma que este pode fazer uso de uma função de transferência própria.
* As funções de transferências geradas devem ser arquivos de extensão “tf1d” e seguir o seguinte formato:
  + <comentário>
  + <algarismo zero>
  + <quantidade de atributos de cor>
  + <primeiro atributo de cor>

⋮

* + <último atributo de cor>
  + <quantidade de atributos de opacidade>
  + <primeiro atributo de opacidade >

⋮

* + <último atributo de opacidade >
* Cada atributo de cor deve ser especificado por 4 números espaçados variando de 0 a 255, onde os 3 primeiros representam a cor em rgb e o último representa o valor de intensidade a qual a cor está associada.
* Cada atributo de opacidade deve ser especificado por 2 números espaçados: o valor da opacidade e o valor de intensidade a qual a opacidade está associada. A opacidade deve variar de 0 a 1 e a intensidade de 0 a 255.

1. Desenvolvimento

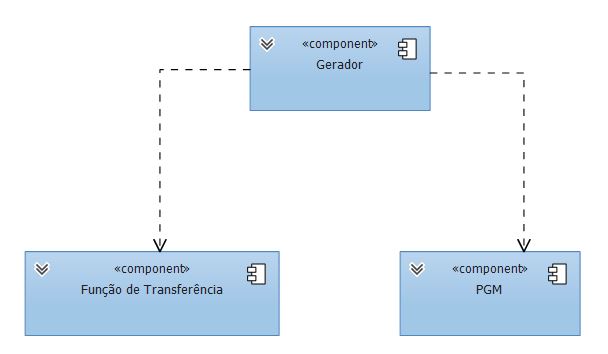
Para a realização deste projeto, foi decidido utilizar o Visual Studio como ambiente de desenvolvimento e C++ como linguagem de programação. O programa é constituído de três projetos internos a uma só solução do Visual Studio:

* ATFGeneratorApp – Projeto de aplicação em console. Contém código do visualizador volumétrico e o código desenvolvido neste projeto.
* ATFGeneratorLib – Projeto de biblioteca estática. O diretório deste projeto é o mesmo do projeto de aplicação. Assim, o mesmo código é compilado em uma biblioteca, para ser usado pelo projeto de testes.
* ATFGeneratorTest – Projeto de testes em console. Contém todos os casos de teste separados por módulo. Cada módulo possui um arquivo de teste.

Na pasta raiz do projeto encontram-se, além deste documento, o manual do usuário, arquivos da solução do Visual Studio e 5 pastas: ATFGenerator, ATFGeneratorTest, bin, include e lib.

1. Arquitetura

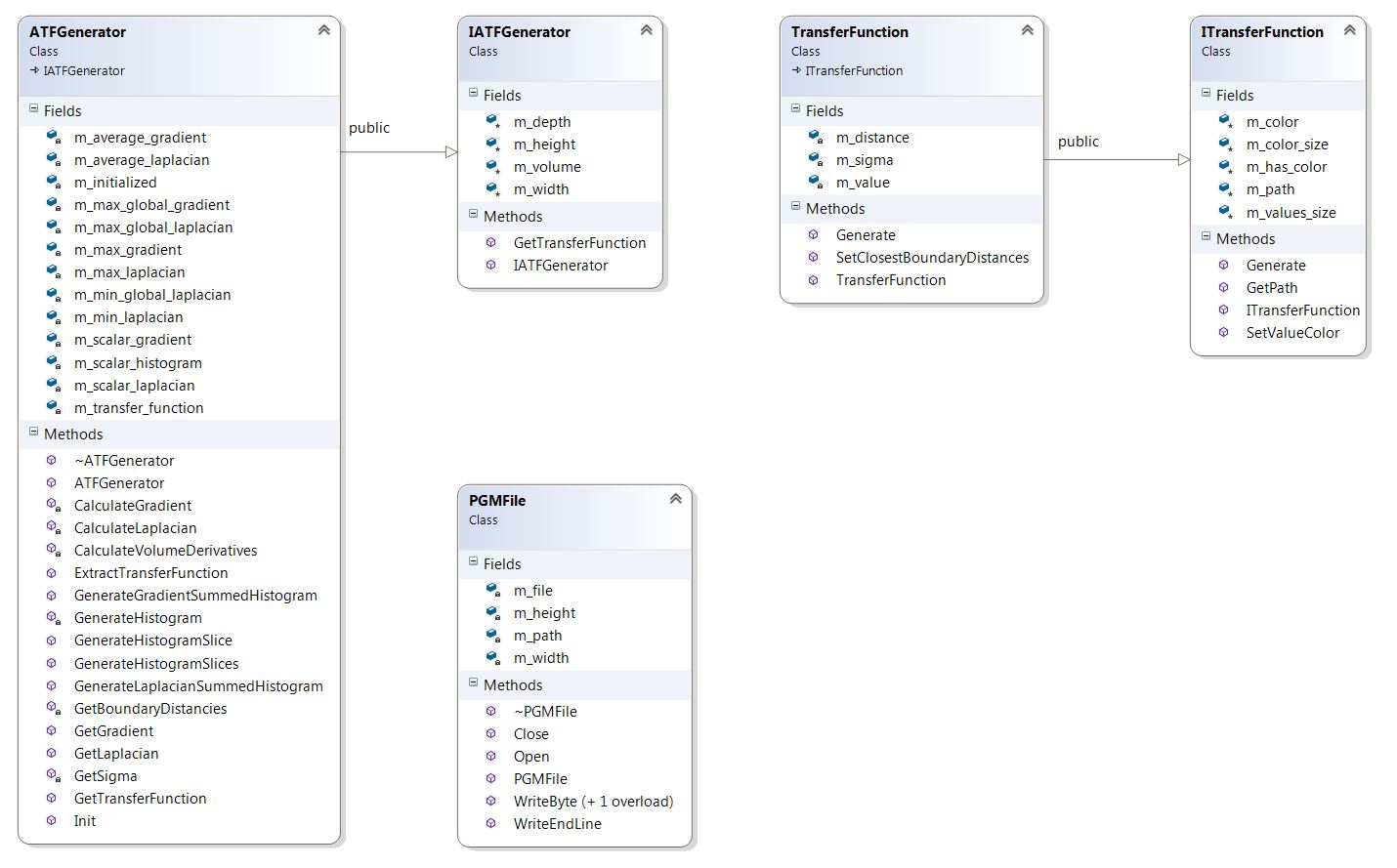
O projeto foi desenvolvido em 3 módulos: Gerador, Função de Transferência e PGM. O módulo PGM fornece uma interface para gerar imagens no formato “PGM”. O módulo Função de Transferência fornece uma interface para gerar funções de transferência. E o módulo Gerador é responsável por analisar o volume de entrada e a partir dele extrair uma função de transferência. A **Figura 1** ilustra a relação de dependência entre os módulos.



**Figura 1:** Diagrama de componentes do projeto.

O programa foi projetado de forma a permitir a implementação de outros métodos que geram funções de transferência automaticamente. Para atingir esse objetivo, os módulos Gerador e Função de Transferência implementam interfaces que especificam a troca de informações entre o visualizador e o programa. Desta forma, novas técnicas podem ser utilizadas a partir de novas implementações das mesmas interfaces.

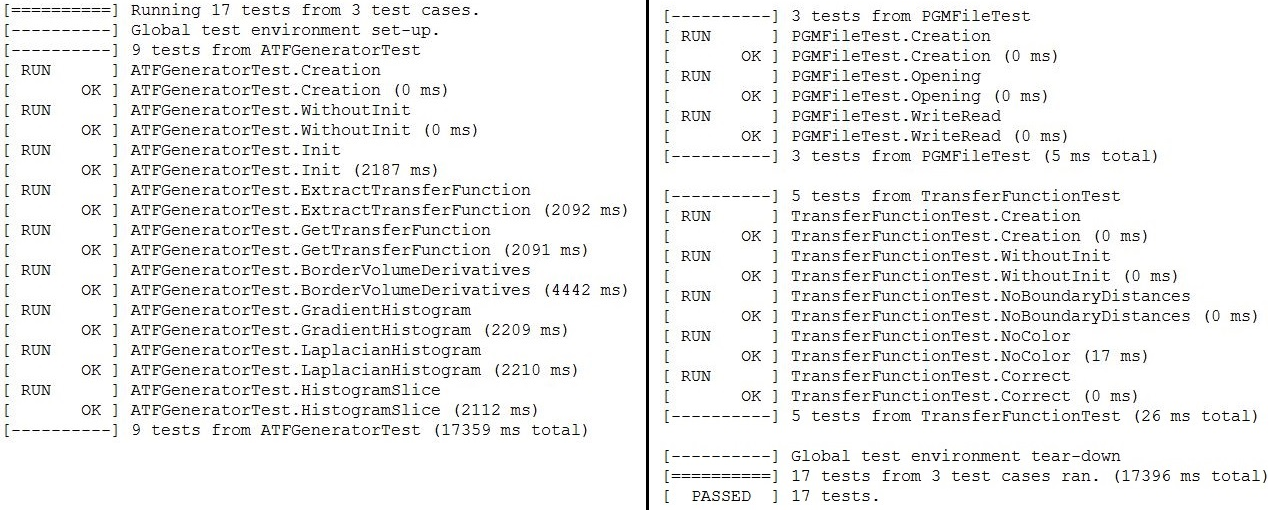
O módulo Gerador é implementado nas classes IATFGenerator e ATFGenerator, o módulo Função de Transferência é implementado nas classes ITransferFunction e TransferFunction, e o módulo PGM é implementado na classe PGMFile. A **Figura 2** mostra o relacionamento e a composição das classes.



**Figura 2:** Diagrama de classes do projeto.

1. Testes

Foram realizados testes unitários com o auxílio da biblioteca de GTest. O relatório de execução de todos os testes está indicado na **Figura 3** e exibe os resultados parciais e finais para cada caso de teste, bem como o resultado geral dos testes.



**Figura 3:** Resultado dos testes unitários.

1. Resultados