

1

Introdução

Motivação: Complexos celulares são amplamente utilizados para representar objetos multi-dimensionais em vários tipos de aplicações. Em particular, complexos simpliciais se destacam pois suas propriedades combinatórias os fazem mais fáceis de ser entendidos, representados e manipulados do que complexos celulares gerais.

Com frequência, sistemas de modelagem geométrica ou de visualização científica precisam lidar com malhas simpliciais extremamente grandes, e com isso, o estudo na área de estruturas de dados topológicas se torna fundamental para otimizar a relação entre a flexibilidade, generalidade e eficiência computacional (tanto no uso de memória quanto na performance) da representação de modelos geométricos.

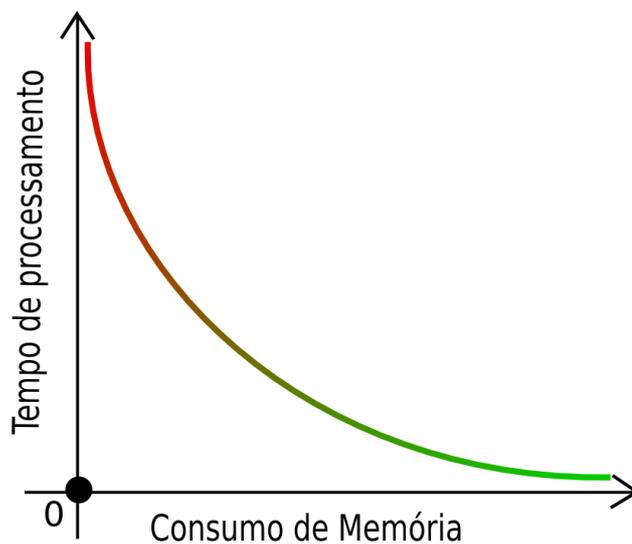


Figura 1.1: Gráfico *performance X memória*.

Ainda, é natural que exista uma relação direta entre o uso de memória e a eficiência da representação (figura 1.1). Quanto mais informação é armazenada, mais memória é requerida, e mais eficiente é a busca das informações topológicas desejadas. Infelizmente, memória é uma limitação física de siste-

mas computacionais e muito embora a performance seja também um aspecto de extrema importância, o tempo é, muitas vezes, uma limitação tolerável.

Por exemplo, quando dobramos a quantidade de dados de entrada que desejamos processar é aceitável que se espere o dobro do tempo do processamento anterior (até mesmo pouco mais do que isso) para que se obtenha o resultado. Entretanto, dobrar a quantidade de dados de entrada pode fazer com que a estrutura de dados não possa ser totalmente armazenada na memória de acesso direto disponível no sistema (RAM), e assim precisaríamos trabalhar com memória virtual para armazenar os dados, o que certamente causaria uma limitação severa que acarretaria na perda significativa de performance (disk thrashing).

Contribuição: Observando este cenário, propomos neste trabalho a implementação de duas estruturas de dados, uma para superfícies e outra para 3-variedades, chamadas de CHE (*Compact Half-Edge*), e CHF (*Compact Half-Face*) (Lage *et al.* 2005) respectivamente. Tais estruturas têm como principal característica o balanceamento entre o uso de memória e a eficiência da representação topológica do modelo.

Em outros termos, apresentaremos duas estruturas de dados *escalonáveis*, no sentido que estas podem utilizar, desde que haja disponibilidade física, uma quantidade adicional de memória para obter melhor desempenho no acesso às informações topológicas, ou por outro lado diminuir a performance da estrutura com o objetivo de aumentar a complexidade dos modelos que podem ser representados.

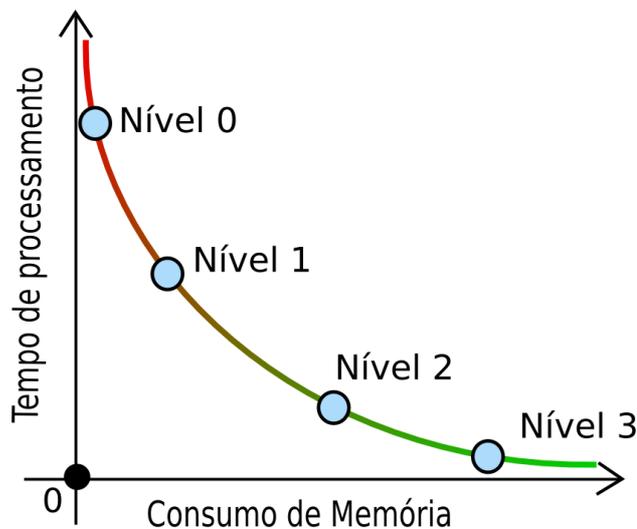


Figura 1.2: Níveis da CHE e da CHF no gráfico *performance X memória*.

Tanto a CHE quanto a CHF são compostas de quatro níveis (figura 1.2), cada um adicionando novas informações ao nível anterior, com o objetivo de

tornar a estrutura mais eficiente na obtenção de informações topológicas. Fica claro no entanto que, a cada novo nível, é preciso alocar uma quantidade maior de memória que no nível anterior.

Na prática, os níveis da estrutura de dados são implícitos ao programador através do uso de herança virtual. *Heranças virtuais* são um recurso da linguagem de programação C++ e, da programação orientada a objetos, que faz com que o programador não se preocupe com qual nível da estrutura ele está trabalhando. O compilador utiliza a seguinte técnica: se um objeto pelo menos uma função virtual, um ponteiro oculto, denominado *v-pointer* é criado. Tal ponteiro aponta para uma tabela global denominada *v-table*. O compilador cria uma *v-table* para cada classe que possui uma ou mais funções virtuais. Na *v-table* são armazenados os endereços de cada função virtual. Durante a execução de uma função virtual, o compilador segue o *v-pointer* do objeto para acessar o elemento apropriado da *v-table* da classe. Assim obtemos uma interface única para cada função da estrutura independente do nível em que estamos trabalhando.

Visão Geral: Dividimos esta dissertação da seguinte forma: no capítulo 2, definimos alguns objetos combinatórios que compõem a base teórica que precisaremos no decorrer do trabalho. No capítulo 3 fazemos uma breve revisão dos trabalhos anteriores em estrutura de dados. No capítulo 4, descrevemos nossa proposta de estrutura de dados escalonável para superfícies, a CHE. No capítulo 5 propomos uma extensão da CHE para variedades de dimensão 3, chamada CHF. No capítulo 6 comparamos o consumo de memória da CHE e da CHF com o de estruturas clássicas da literatura. Por fim, no capítulo 7 apresentamos algumas conclusões e propomos trabalhos futuros.