Filtragem em Espaços de Alta Dimensionalidade para processamento de Imagens e Vídeos

Marcos H. Backes, Manuel M. Oliveira

Universidade Federal do Rio Grande do Sul

1. Introdução

A otimização de programas não é uma tarefa simples. A eficiência de um programa depende da complexidade do algoritmo, da velocidade do hardware e do uso eficiente deste. Trata-se de um problema de otimização não trivial que busca balancear o paralelismo das operações, a localidade no acesso aos dados, e evitar trabalho redundante. A linguagem de programação Halide [2] se propõe a facilitar o processo de otimização de programas para processamento de imagens, aumentando assim a produtividade dos programadores e o desempenho das aplicações desenvolvidas. O objetivo desse trabalho consistiu em explorar os recursos da linguagem Halide e implementar a técnica de filtragem de imagens baseada na *Domain Transform* [1].

2. Halide

Halide [2] é uma nova linguagem de programação voltada para processamento de imagens. Seu principal objetivo é facilitar a geração de código otimizado para diferentes tipos de arquiteturas, incluindo CPUs, GPUs e dispositivos móveis, como *smartphones*. Para isso, a linguagem divide o programa em duas partes: (1) algoritmo, que define quais operações devem ser realizadas; e (2) *schedule*, que define a ordem de execução e a arquitetura utilizada. Assim, a linguagem facilita a busca pela melhor implementação, permitindo testar várias alternativas de *schedules* para um mesmo algoritmo.

3. Domain Transform

A *Domain Transform* [1] é uma técnica de filtragem para suavização de imagens com preservação de arestas, que se caracteriza por sua eficiência e facilidade de paralelização, podendo ser aplicada a imagens e vídeos em tempo real. Além disso, possui grande versatilidade, sendo utilizada em diversas aplicações, como realce de detalhes, estilização, e recoloração, entre outras.

Figura 1: Exemplos de aplicações da *Domain Transform*. Da esquerda para a direita: imagem original, suavização com preservação de arestas, realce de detalhes e estilização.

4. Resultados

Três filtros associados à *Domain Transform* - *recursive filter* (RF), *normalized convolution* (NC) e *interpolated convolution* (IC) - foram implementados utilizando a linguagem Halide. Foram realizadas comparações entre os tempos de execução da implementação original [1] e em Halide.

Tabela 1: Comparação do tempo de execução entre as implementações dos filtros original (C++, OpenMP) e em Halide para CPU, para imagens de 1 e 10 megapixels (MP).

	1 MP		10 MP	
Filtro	**Original**	**Halide**	**Original**	**Halide**
RF	14ms	28ms	207ms	301ms
NC	25ms	31ms	218ms	379ms
IC	-	68ms	-	782ms

5. Conclusão

O estudo sobre Halide permitiu um entendimento sobre os processos associados à otimização de código e sua interdependência com as várias arquiteturas de hardware. O aprendizado sobre a técnica de filtragem de imagens com preservação de arestas foi uma experiência importante, oferecendo uma ótima exposição à esta sub-área de processamento de imagens e vídeos.

Referências

[1] E. S. L. Gastal and M. M. Oliveira. Domain transform for edge-aware image and video processing. *ACM Trans. Graph.*, 30(4):69:1–69:12, Julho 2011.

[2] J. Ragan-Kelley, A. Adams, S. Paris, M. Levoy, S. Amarasinghe, and F. Durand. Decoupling algorithms from schedules for easy optimization of image processing pipelines. *ACM Trans. Graph.*, 31(4):32:1–32:12, Julho 2012.

Agradecimentos

CNPq Processo Nº 134657/2017-9.