

UNIVERSIDADE FEDERAL DE ALAGOAS
INSTITUTO DE COMPUTAÇÃO

DANILO FERNANDES COSTA
EDUARDO GOMES DOS SANTOS
MARCELO CAVALCANTE LIMA
VICTOR MAFRA DE HOLANDA FERRAZ
WILLIENY BARBOSA DE MAGALHÃES

**Implementação em FPGA de Graph Cuts para Visão Estéreo utilizando
múltiplas pipelines e memória externa com mecanismo on-chip cache**

Maceió - AL
2018

DANILO FERNANDES COSTA
EDUARDO GOMES DOS SANTOS
MARCELO CAVALCANTE LIMA
VICTOR MAFRA DE HOLANDA FERRAZ
WILLIENY BARBOSA DE MAGALHÃES

**Implementação em FPGA de Graph Cuts para Visão Estéreo utilizando
múltiplas pipelines e memória externa com mecanismo on-chip cache**

Projeto de Pesquisa apresentado à disciplina de Metodologia de Pesquisa e do Trabalho Científico do curso de Engenharia da Computação da Universidade Federal de Alagoas

SUMÁRIO

1	INTRODUÇÃO	3
1.1	Descrição do Problema de Pesquisa	3
1.2	Justificativa da Pesquisa	3
1.3	Hipóteses (ou Pressupostos) da Pesquisa	4
2	OBJETIVOS DE PESQUISA	4
2.1	Objetivo Geral da Pesquisa	4
2.2	Objetivos Específicos da Pesquisa	4
3	FUNDAMENTAÇÃO TEÓRICA	4
3.1	Field Programmable Gate Array (FPGA)	4
3.2	<i>Graph Cuts</i>	5
3.3	Algoritmo push-relabel	5
4	METODOLOGIA DA PESQUISA	6
4.1	Descrição da Metodologia da Pesquisa	6
	REFERÊNCIAS	7

1 INTRODUÇÃO

1.1 Descrição do Problema de Pesquisa

A análise de dados representados como grafos é amplamente difundida e possui aplicações que se estendem desde redes sociais a processamento de imagens. Uma análise popular é o corte do grafo a fim de formar subgrafos disjuntos que possam representar, por exemplo, objetos de interesse e planos de fundo em imagens (KUO et al., 2015).

Uma ferramenta de otimização poderosa que tem se popularizado na realização desse tipo de análise é o Graph Cuts, a qual tem encontrado uma grande variedade de aplicações em diversos ramos da Visão Computacional, dentre os quais podemos destacar a Visão Estéreo (VINEET; NARAYANAN, 2008).

Visão Estéreo consiste basicamente na reconstrução da informação 3D de uma cena a partir de imagens da mesma obtidas simultaneamente por duas câmeras (VALSARAJ et al., 2016). Essa informação 3D pode ser utilizada para construir um grafo de grade 3D, onde os nós representam os pontos de superfícies hipotéticas e as arestas, as probabilidades da existência daquelas. Consequentemente, a extração de superfícies pode ser feita por meio de Graph Cuts (KAMASAKA; SHIBATA; OGURI, 2017).

Entretanto, todo esse processo tem um custo computacional alto para processadores, resultando em um grande obstáculo para aplicações que almejam tempo-real, como a robótica. No entanto, pode-se obter ganho considerável de eficiência realizando sua implementação em FPGA, pois, além de poder ser configurado a nível de hardware para acomodar a aplicação, provê um ambiente para construção de uma arquitetura de hardware paralela (VALSARAJ et al., 2016).

Uma implementação em tais moldes foi proposta por Kamasaka, Shibata e Oguri (2017), no entanto houve uma grande limitação no tamanho do grafo a ser processado, sendo este restrito a 129x129x16 nós. Logo, é relevante uma implementação que propicie um processamento de grafos consideravelmente maiores e, consequentemente, a extração e análise de grandes superfícies a partir de imagens.

1.2 Justificativa da Pesquisa

Acurácia e processamento em tempo-real são requisitos essenciais para muitas aplicações de Visão Estéreo, tais como os carros autônomos, construção de modelos 3D, realidade aumentada, navegação e manipulação robótica (VALSARAJ et al., 2016).

Partindo desse pressuposto e da relevância da obtenção de superfícies que representem objetos do ambiente no qual o sistema está imerso, é justificável uma implementação com tal finalidade em um dispositivo tão propenso a otimizações como a FPGA de um método acurado como o Graph Cuts, visando maior escalabilidade.

1.3 Hipóteses (ou Pressupostos) da Pesquisa

Será necessário o desenvolvimento de um código em uma linguagem de descrição de hardware específica para a FPGA, visando técnicas que melhor construam uma arquitetura de hardware paralela e manipulem de maneira otimizada possíveis recursos de memória para armazenamento escalável dos grafos.

2 OBJETIVOS DE PESQUISA

2.1 Objetivo Geral da Pesquisa

Desenvolver uma implementação em FPGA para Visão Estéreo utilizando Graph Cuts para a obtenção de superfícies que venham a representar objetos do ambiente com uma maior escalabilidade em relação ao tamanho destas e performance otimizada, a fim de adequar a implementação para aplicações de tempo-real.

2.2 Objetivos Específicos da Pesquisa

Implementar o método Graph Cuts em FPGA para a extração de superfícies contidas em grafos, que representam cenários, gerados por meio de Visão Estéreo, utilizando memória externa para o armazenamento desses. Além disso, será feito o uso de múltiplas pipelines e desenvolvido um mecanismo on-chip cache, a fim de prover uma melhor performance mediante os atrasos resultantes do acesso à memória externa.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 Field Programmable Gate Array (FPGA)

FPGA é um chip composto por blocos lógicos que permitem a implementação de um código em linguagem de descrição de hardware. Nas bordas de um chip FPGA estão os blocos de entrada e saída (IOs), eles fazem a comunicação entre o usuário e os blocos que estão no interior do chip. Esses blocos se comunicam entre si através de uma rede de interconexão (MONMASSON; CIRSTEIA, 2007).

Os blocos lógicos são controlados por células de memória, que possuem várias tecnologias diferentes, entre elas FLASH e SRAM são as que possuem a mesma flexibilidade dos microprocessadores por serem reprogramáveis. Atualmente há arquiteturas com uma complexidade maior, com blocos dedicados como RAM, DSP, núcleos de processadores embutidos e entre outros (KAMASAKA; SHIBATA; OGURI, 2017).

3.2 Graph Cuts

Considere o grafo direcional $G = (V, E)$, onde V é o conjunto de vértices e E , o de arestas; e dois nós terminais s (*source*) e t (*sink*) pertencentes a V . Considere também que cada aresta $(u, v) \in E$, onde $u, v \in V$, está associada a um valor $c(u, v)$ não negativo denominado capacidade da aresta (VEKSLER, 2008).

Define-se um corte $C \subset E$ nesse grafo como um subconjunto de arestas que se for removido de G , irá dividi-lo em dois subgrafos cujos conjuntos de vértices são subconjuntos disjuntos de V e são representados por S e T , de modo que $s \in S$ e $t \in T$. Posto isso, define-se como a capacidade de um corte C como sendo o valor dado por

$$|C| = \sum_{(u,v) \in C} c(u, v). \quad (1)$$

Portanto, *Graph Cuts* é o método que visa encontrar um corte C em um dado grafo cuja capacidade seja mínima, o qual é denominado *min-cut* (VEKSLER, 2008).

O problema de encontrar o *min-cut*

O problema fluxo-máximo é definido como o problema para encontrar o fluxo máximo de uma rede de fluxos, a qual é representada em um grafo direcionado com capacidade em suas arestas. O fluxo de u para v é definido por $f(u, v)$, onde $0 \leq f(u, v) \leq c(u, v)$ (KAMASAKA; SHIBATA; OGURI, 2017).

O teorema fluxo-máximo/corte-mínimo diz que a quantidade de fluxo máximo é igual a capacidade de corte mínimo, dessa forma, podemos substituir o problema de encontrar o corte mínimo pelo problema de encontrar o fluxo máximo de uma rede (VEKSLER, 2008).

Existem dois métodos principais para encontrar o fluxo máximo de uma rede, o método Ford-Fulkerson e o método Push-relabel. Esses algoritmos usam o conceito de rede residual, que é rede criada a partir da rede de fluxo, mas com capacidades remanescentes em seus vértices, tal capacidade remanescente é dada por:

$$c_f(u, v) = \begin{cases} c(u, v) - f(u, v) & (u, v) \in E \\ f(v, u) & (v, u) \in E \\ 0 & \text{de outra forma} \end{cases}$$

3.3 Algoritmo push-relabel

Os algoritmos push-relabel agem de uma forma mais localizada que o método Ford-Fulkerson. Em vez de examinar toda a rede residual com o objetivo de encontrar um caminho aumentador, algoritmos push-relabel atuam em um vértice por vez, examinando somente os vizinhos do vértice na rede residual. Além disso, diferentemente do método

Ford-Fulkerson, os algoritmos push-relabel não mantêm a propriedade de conservar o fluxo e aumentá-lo iterativamente durante toda a sua execução.(CORMEN et al., 2009)

Entretanto, eles mantêm um pré-fluxo que flui de um nó superior para um nó inferior. Esse pré-fluxo pode ter uma quantidade maior para entrada de um nó do que para a saída de um nó. Assim, existe a possibilidade de que os fluxos que não podem fluir de um nó sejam acumulados.(KAMASAKA; SHIBATA; OGURI, 2017)

Esse fluxo acumulado é chamado de excesso de fluxo e é definido por

$$e(u) = \sum_{v \in V} g(u, v) \geq 0, \forall v \in V - \{s\}$$

Um nó v com $e(v) > 0$ é chamado de nó ativo. No método push-relabel, o fluxo máximo é obtido aplicando iterativamente a operação push e a operação relabel aos nós ativos. (KAMASAKA; SHIBATA; OGURI, 2017)

- **PUSH**(u, v): Se o nó u estiver transbordando, a sua altura $h(u)$ for maior do que a altura $h(v)$, no qual v é um dos seus nós vizinho, e $c_f(u, v) > 0$, um fluxo excedente é empurrado de u para v o máximo possível.
- **RELABEL**(u): se o nó u estiver transbordando e a altura $h(u)$ for menor ou igual do que todos os seus vizinhos que podem aceitar um fluxo, então u será remarcado, a sua altura $h(u)$ é aumentada em 1 da altura mais baixa entre os nós vizinhos.

Repetindo essas duas operações na medida em que existem nós ativos, o fluxo máximo será obtido.

4 METODOLOGIA DA PESQUISA

4.1 Descrição da Metodologia da Pesquisa

Inicialmente a ideia é usar um FPGA de modelo cyclone IV EP4CE6 da fabricante Altera, que conta com 270kbits de memória interna e 200MHz de frequência operacional máxima.

Depois de feita a implementação da memória interna e elaborado o novo algoritmo, serão feitos testes em diferentes ambientes e os dados serão coletados para análise para fazer os ajustes necessários.

REFERÊNCIAS

- CORMEN, T. H. et al. *Introduction to Algorithms, Third Edition*. 3rd. ed. [S.l.]: The MIT Press, 2009. ISBN 0262033844, 9780262033848.
- KAMASAKA, R.; SHIBATA, Y.; OGURI, K. Fpga implementation of a graph cut algorithm for stereo vision. In: *Proceedings of the 8th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies*. New York, NY, USA: ACM, 2017. (HEART2017), p. 14:1–14:6. ISBN 978-1-4503-5316-8. Disponível em: <http://doi.acm.org/10.1145/3120895.3120907>.
- KUO, C.-T. et al. Unified and contrasting cuts in multiple graphs: Application to medical imaging segmentation. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2015. (KDD '15), p. 617–626. ISBN 978-1-4503-3664-2. Disponível em: <http://doi.acm.org/10.1145/2783258.2783318>.
- MONMASSON, E.; CIRSTEA, M. N. Fpga design methodology for industrial control systems—a review. *IEEE transactions on industrial electronics*, IEEE, v. 54, n. 4, p. 1824–1842, 2007.
- VALSARAJ, A. et al. Stereo vision system implemented on fpga. *Procedia Technology*, v. 24, p. 1105 – 1112, 2016. ISSN 2212-0173. International Conference on Emerging Trends in Engineering, Science and Technology (ICETEST - 2015). Disponível em: <http://www.sciencedirect.com/science/article/pii/S2212017316303346>.
- VEKSLER, O. Star shape prior for graph-cut image segmentation. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2008. p. 454–467.
- VINEET, V.; NARAYANAN, P. J. Cuda cuts: Fast graph cuts on the gpu. In: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. [S.l.: s.n.], 2008. p. 1–8. ISSN 2160-7508.