

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

## **Grupo de Pesquisa:**

Danilo Fernandes Costas  
Eduardo Gomes dos Santos  
Marcelo Cavalcante Lima  
Victor Mafra de Holanda Ferraz  
Willieny Barbosa de Magalhães



# Sumário

1.	Introdução.....	3
1.1.	Descrição do Problema de Pesquisa.....	3
1.2.	Justificativa da Pesquisa.....	4
1.3.	Hipóteses da Pesquisa.....	5
2.	Objetivos da Pesquisa.....	6
2.1.	Objetivo Geral da Pesquisa.....	6
2.2.	Objetivos Específicos da Pesquisa.....	7
3.	Fundamentação Teórica.....	8
3.1.	Field Programmable Gate Array (FPGA).....	8
3.2.	Graph Cut.....	9
3.3.	Algoritmo Push-Relabel.....	10
4.	Metodologia da Pesquisa.....	11
4.1.	Descrição da metodologia da pesquisa.....	11
5.	Cronograma.....	12
5.1.	Cronograma do projeto de pesquisa.....	12
6.	Anexos.....	13
7.	Referências.....	14



# 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 o objeto de interesse (KUO et al., 2015).

O *Graph Cuts* é uma ferramenta de otimização que vem se popularizando nesta área, tendo várias aplicações para os ramos da Visão Computacional, principalmente para a Visão Estéreo (reconstrução de uma informação 3D a partir das imagens capturadas por 2 câmeras diferentes) (VINEET; NARAYANAN, 2008).

Porém, todo esse processo tem um custo computacional muito elevado para processadores, sendo um grande obstáculo para aplicações que buscam o tempo-real, como a robótica. Com a utilização de FPGA, podemos ter ganhos consideráveis de eficiência, pois permite a configuração em nível de *hardware* (VALSARAJ et al., 2016).



# 1. Introdução

## 1.2 Justificativa da Pesquisa:

Precisão e processamento em tempo-real são essenciais para várias aplicações da Visão Estéreo, partindo desse pressuposto e da relevância da obtenção de superfícies que representam objetos do ambiente no qual o sistema está inserido, é justificável uma implementação do *Graph Cuts* com esta finalidade em FPGA buscando maior escalabilidade, já que o mesmo é muito propenso a otimização.



# 1. Introdução

## 1.3 Hipóteses 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 da 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. Objetivos da Pesquisa

### 2.1 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.
- Será desenvolvido um mecanismo *on-chip cache* e também será feito o uso de múltiplas *pipelines*, a fim de prover uma melhor performance diante dos 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* (HDL). Tais blocos 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 pois são reprogramáveis (MONMASSON; CIRSTEIA, 2007).





## 3. Fundamentação Teórica

### 3.2 Graph Cut:

*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*. Contudo, encontrar o *min-cut* de um grafo consiste em um problema o qual, segundo o Teorema de Ford e Fulkerson, é equivalente ao problema do *max-flow*, para o qual já são conhecidos algoritmos que o solucione (FORD; FULKERSON, 1956).

O problema do *max-flow* 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. Dentre os algoritmos existentes para solucionar o problema anterior, os principais são o de Ford-Fulkerson e o *Push-Relabel* (KAMASAKA; SHIBATA; OGURI, 2017).



## 3. Fundamentação Teórica

### 3.3 Algoritmo push-relabel:

Os algoritmos *Push-Relabel* atuam em um vértice por vez, examinando somente os vizinhos do vértice na rede residual. Eles mantêm um pré-fluxo que flui de um nó superior para um nó inferior. Esse pré-fluxo pode acabar acumulando e é chamado de excesso de fluxo (CORMEN et al., 2009).

No algoritmo *Push-Relabel*, o fluxo máximo é obtido aplicando iterativamente aos nós ativos as seguintes operações:

- *PUSH*( $u, v$ ): se o nó  $u$  for um nó ativo,  $v$  um de seus vizinhos,  $h(u) > h(v)$  e capacidade residual maior que 0, então o fluxo é empurrado de  $u$  para  $v$  o máximo possível.
- *RELABEL*( $u$ ): se o nó  $u$  for um nó ativo,  $v$  um de seus vizinhos e  $h(u) \leq h(v)$ , então  $u$  será remarcado e sua altura é aumentada em 1 da altura de  $v$ .



## 4. Metodologia da Pesquisa

### 4.1 Descrição da Metodologia da Pesquisa:

Será feita uma análise dos modelos disponíveis de FPGA e avaliação dos tipos de memórias externas disponíveis, dando ênfase à aspectos como memória interna e frequência operacional máxima, a fim de determinar aquela na qual será feita a implementação e os testes.

Feito isto, serão desenvolvidos o mecanismo de *on-chip cache* e o código em VHDL que implemente o algoritmo *Push-Relabel*.

Após a implementação, serão feitos testes em diversos ambientes a fim de coletar dados para análise estatística de aspectos como acurácia, performance, uso de memória e consumo energético.

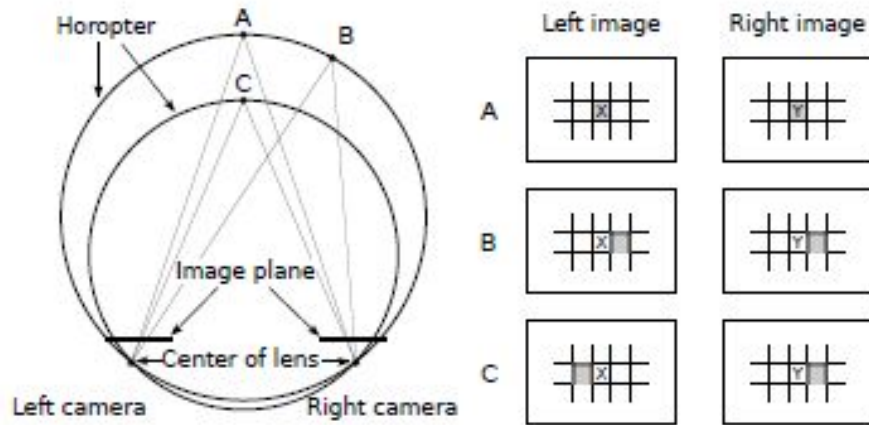
# 5. Cronograma



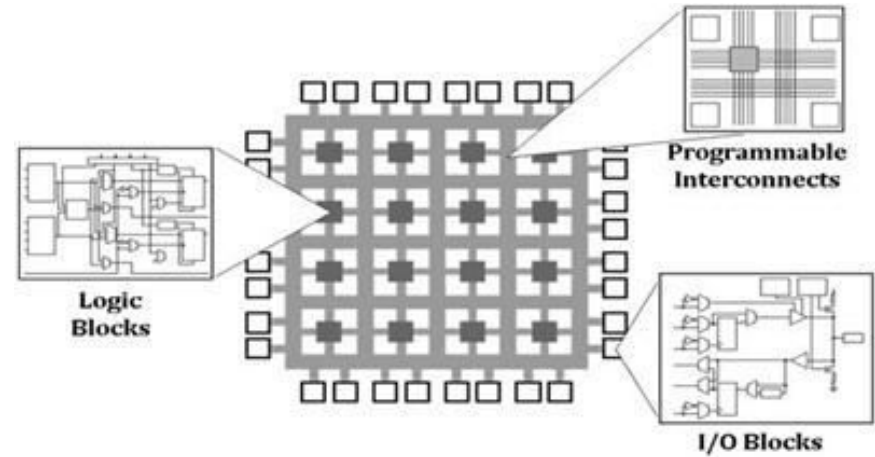
## 5.1 Cronograma do Projeto de Pesquisa:

Atividades	2018							2019				
	JUN	JUL	AGO	SET	OUT	NOV	DEZ	JAN	FEV	MAR	ABR	MAI
Análise dos modelos de FPGA e avaliação dos tipos de memória externa disponíveis												
Desenvolvimento do mecanismo on-chip cache												
Desenvolvimento do código em VHDL												
Estudo contínuo dos recursos de memória												
Testes em diversos ambientes e análise estatística												

## 6. Anexos



Fonte: Kamasaka, Shibata e Oguri (2017)



Fonte: Brown (1992)

## 7. Referências

CORMEN, T. H. et al. Introduction to Algorithms, Third Edition. 3rd. ed. [S.I.]: 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/10.1145/3120895.3120907i>.

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/10.1145/2783258.2783318i>.

## 7. Referências

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/S2212017316303346i>.

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.