PARALELIZAÇÃO DE ALGORITMOS DE TRIMAP UTILIZANDO A ARQUITETURA CUDA

Aluno: Henrique Ferraz de Arruda
Orientador: Prof. Adj. Antônio Carlos Sementille
Co-orientador: Prof. Adj. João Fernando Marar

13 de Novembro de 2012

Cronograma

Técnicas de combinações de imagens, com extração de objetos para montar uma cena final, são muito utilizadas em aplicações que vão desde montagens em fotos até produções cinematográficas.
A partir delas é possível diminuir o custo das produções e melhorar o resultado final.

O *matting* digital é uma técnica para combinação de imagens digitais, onde há a extração dos objetos em primeiro plano de imagens fixas ou uma sequência de imagens (vídeo). Essa técnica tem sido amplamente estudada há mais de 20 anos.
A extração consiste em separar os objetos em primeiro plano do plano de fundo, considerando o nível de transparência dos objetos.

Para que sejam mantidas as transparências da superfície, é utilizado o canal alfa, o qual representa o nível de transparência de cada pixel da imagem. Representa, portanto, a interpolação linear de cores entre as duas camadas (plano de fundo e plano da frente) e varia entre 0 e 1 [Porter e Duff 1984]. A composição das imagens é calculada utilizando a Equação 1.

$$C = \alpha F + (1–\alpha)B \tag{1}$$

O valor de alfa computacionalmente é armazenado em uma variável de 8 bits, dessa forma é necessário fazer uma mudança de escala, na qual o valor máximo passa a ser 255 e deixa de ser real [Smith e Blinn 1996].

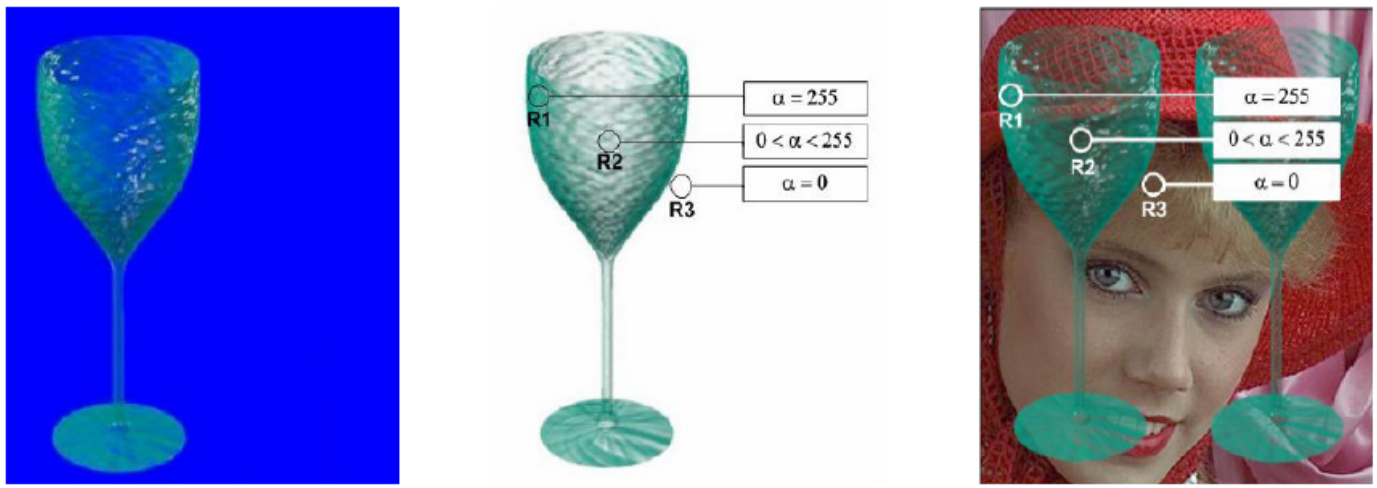


Figura: (a) o objeto à ser extraído, (b)Extração do objeto (c) composição final. Fonte: [Smith e Blinn 1996].

O *trimap* consiste, basicamente, em uma demarcação na imagem dos pontos de certeza, que são *background* e *foreground*, e pontos de incerteza, estes que devem ser classificados posteriormente pelo algoritmo de *matting*. Na maior parte dos casos o *trimap* é feito manualmente.

Figura: Exemplo de *trimap*. Fonte: [Wang e Cohen 2007]

CUDA (*Compute Unified Device Architecture*) é a arquitetura de GPGPU (*General Purpose Graphics Processing Unit*), que é uma placa gráfica para uso geral, desenvolvida pela NVIDIA.
Ela se aproveita da presença de múltiplos processadores e alta banda de memória para realizar operações em grande volume de dados. Sua utilização é adequada para resolver problemas que apresentem paralelismo de dados, em que a mesma instrução é executada paralelamente em dados diferentes [Nvidia 2011].

Obter implementações paralelizadas, baseadas na arquitetura CUDA, de técnicas de geração automáticas de *trimaps*, visando a sua utilização em aplicações que envolvam *matting* digital com restrições de tempo.

- Analisar técnicas de *matting*, com informações extras;
- Analisar as funcionalidades do Kinect pertinentes ao projeto;
- Investigar a utilização da arquitetura CUDA na solução da implementação das técnicas de *matting* digital de imagens e vídeo;
- Implementar as técnicas de geração automática de *trimaps*, segundo a abordagem sequencial (para processadores de uso geral) e paralelizada (para a arquitetura CUDA);
- Comparar o desempenho das implementações para a arquitetura CUDA com as implementações para processadores de uso geral.

Pipeline

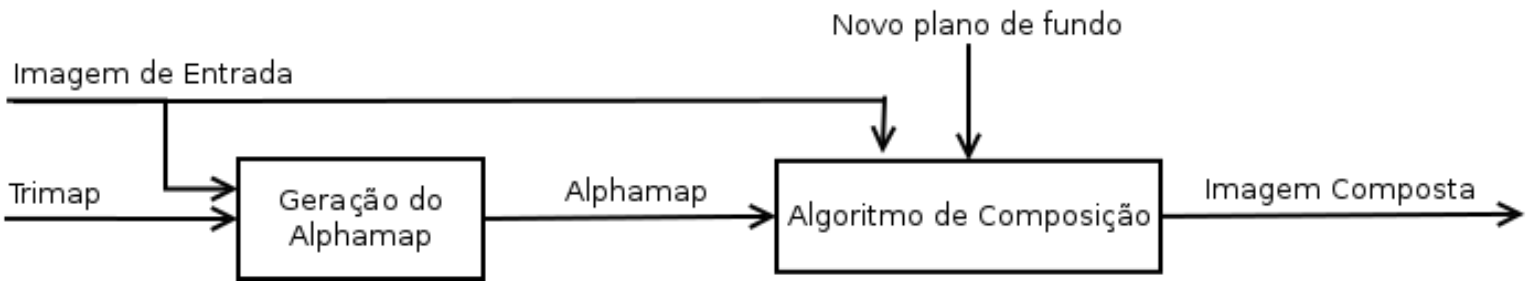


Figura: *Pipeline* do *Matting* Digital.

Chroma Key

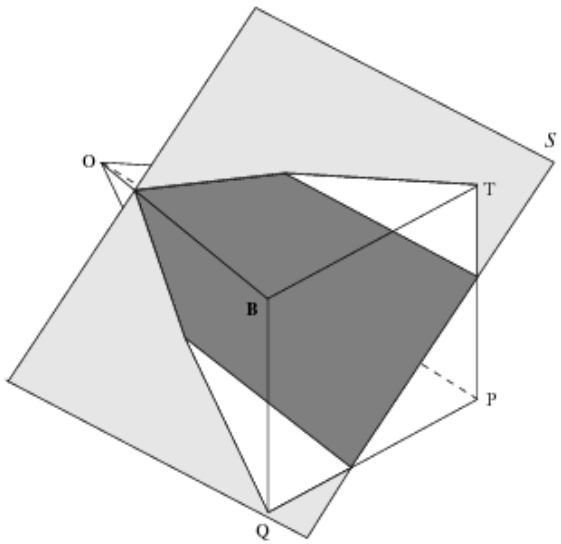


Figura: Cubo RGB com a intersecção do plano S com a pirâmide de inclinação OBTPQ. Fonte: [Bergh e Lalioti 1999].

Pipeline

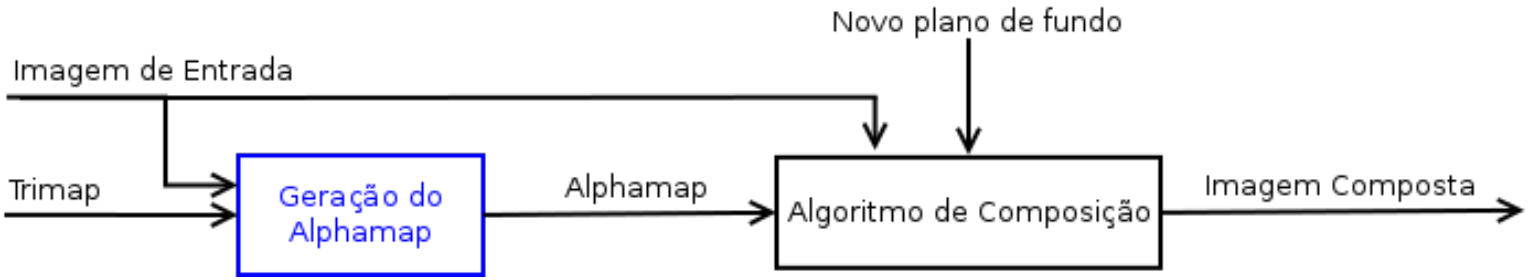


Figura: *Pipeline* do *Matting* Digital.

Para auxiliar nos testes e comparação dos resultados foram implementados o *Bayesian Matting* [Chuang et al. 2001] e o *Robust Matting* [Wang e Cohen 2007], a partir de códigos cedidos pelos autores de [Sun et al. 2006]. Estes programas foram modificados, pois possuíam limitações quanto ao tamanho das imagens e uso de bibliotecas.

Os primeiros métodos foram desenvolvidos com o intuito de gerar o *trimap* automaticamente.

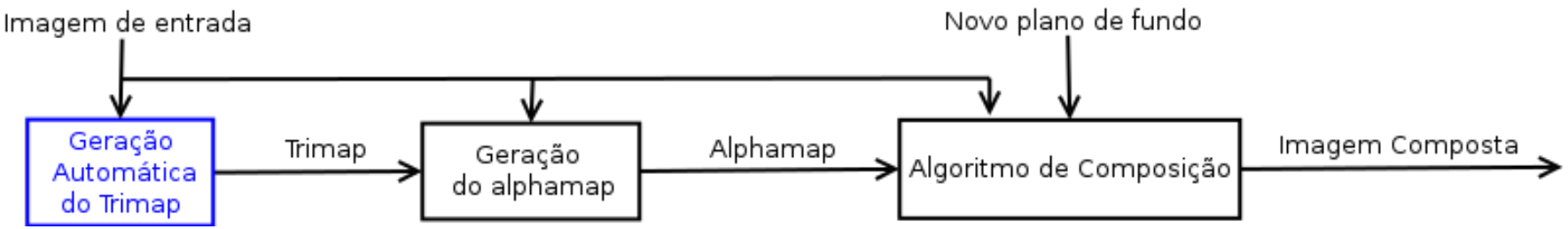


Figura: *Pipeline* do *Matting* Digital.

Automatização do *Trimap* com uso de Kinect e *chroma key*

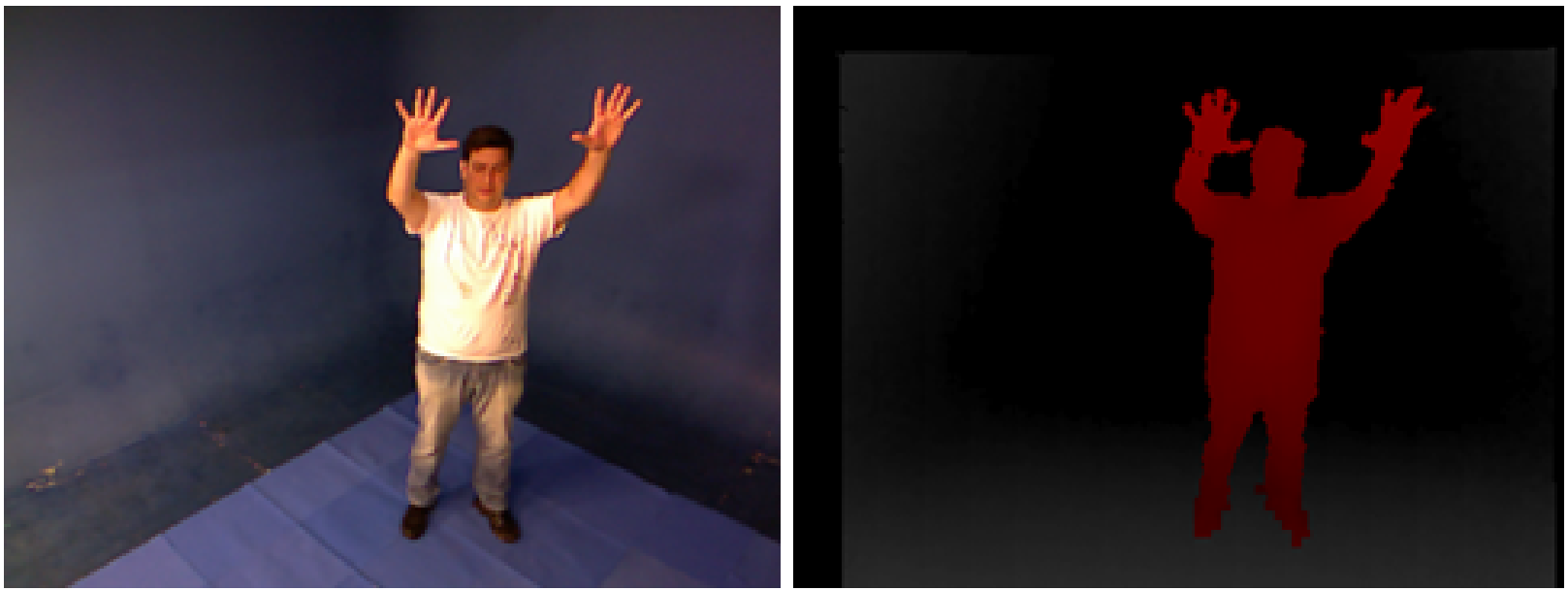


Figura: Imagens de saída do Kinect.

Automatização do *Trimap* com uso de Kinect e *chroma key*

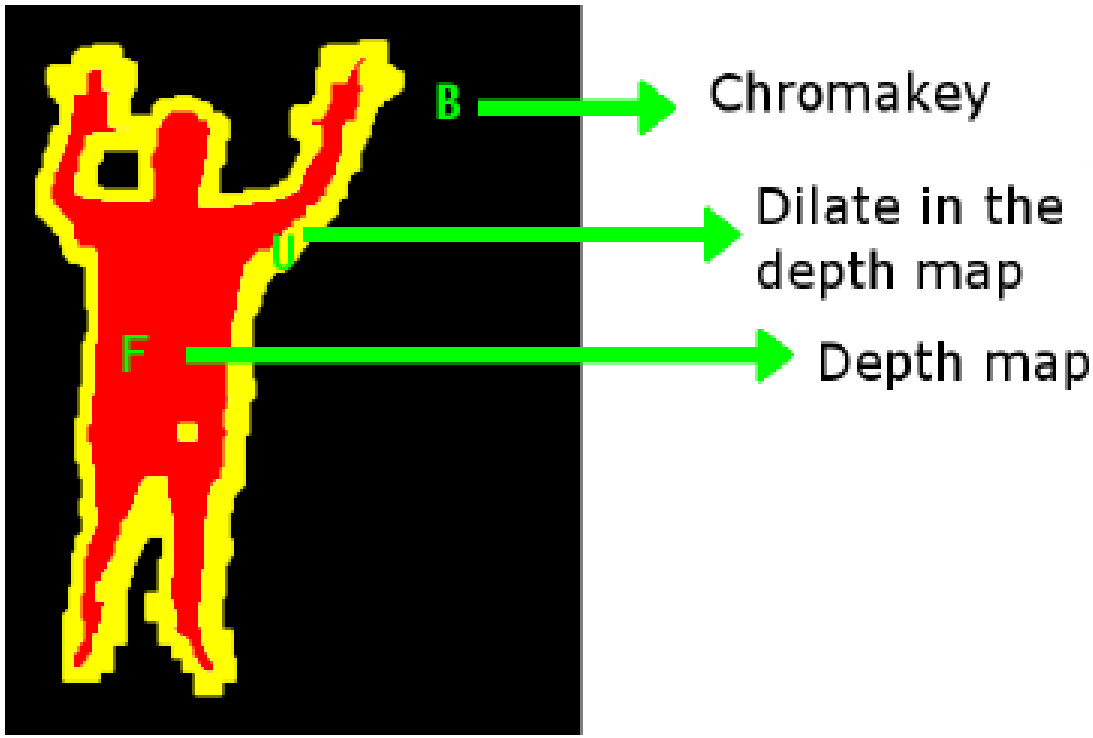


Figura: **Automatização do *Trimap* com uso de Kinect e *chroma key*.**

Automatização do *Trimap* com uso de *chroma key*

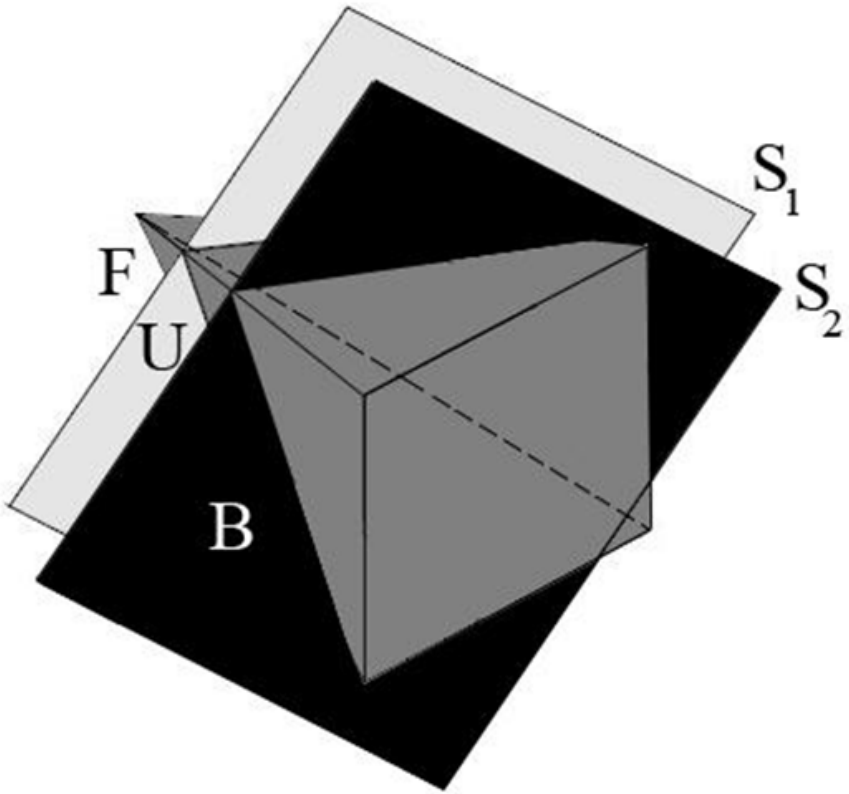


Figura: Automatização do Cubo RGB com a intersecção dos planos S1 e S2.

Geodesic Matting

Foi desenvolvido um método para a geração do *alphamap* com restrições de tempo.

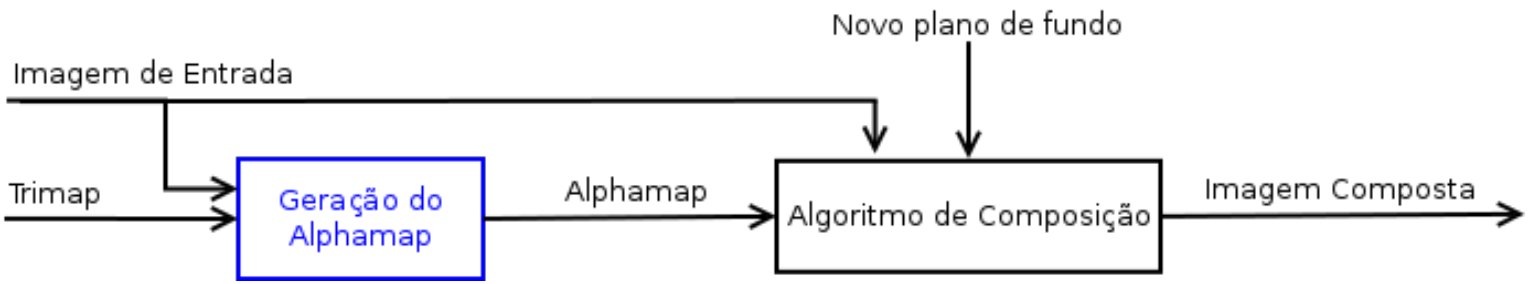


Figura: *Pipeline* do *Matting* Digital.

Geodesic Matting

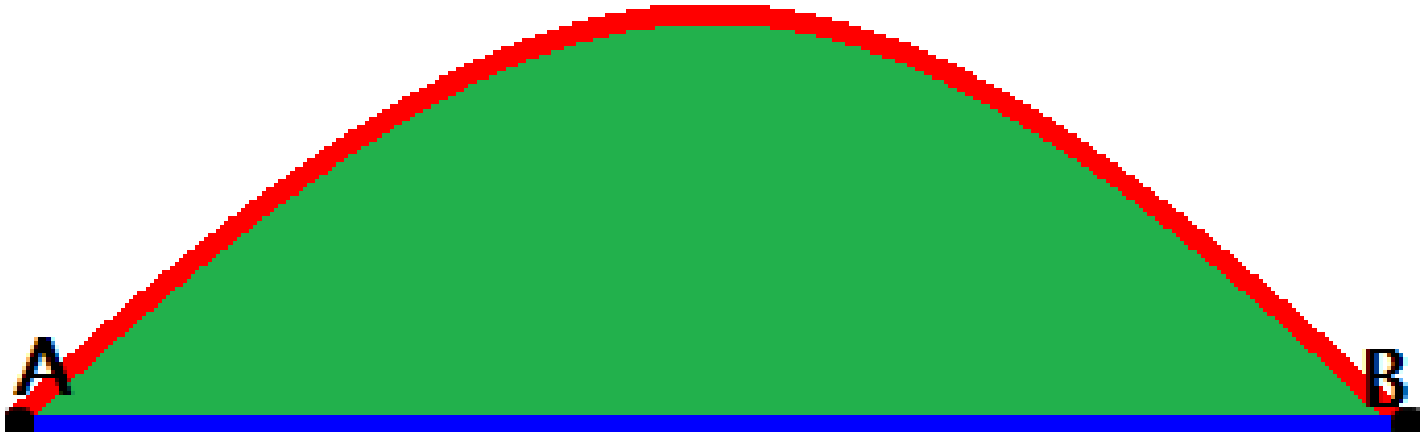
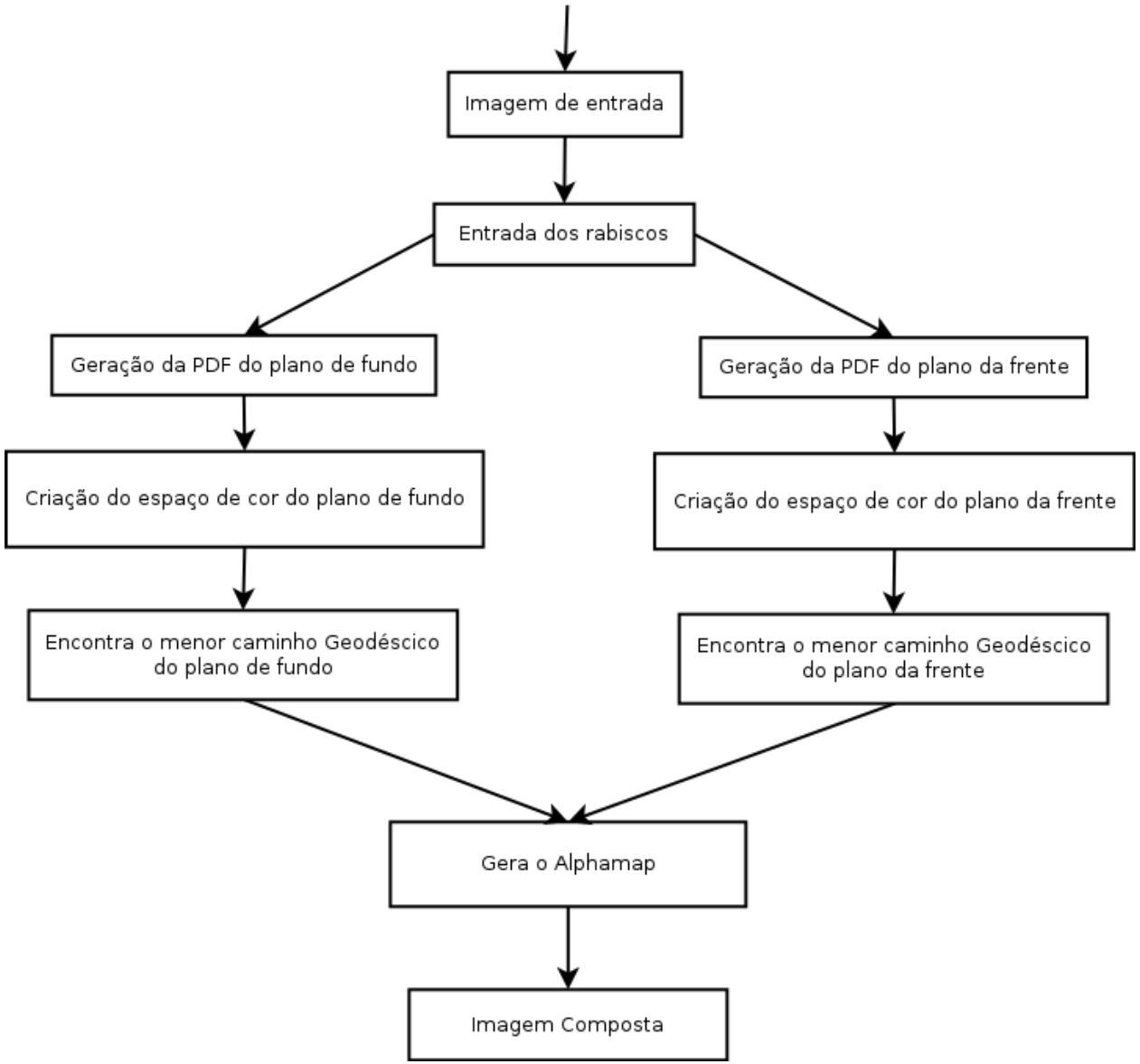


Figura: A e B são os pontos, a distância desenhada em azul é a distância euclidiana e a distância desenhada em vermelho é a distância geodésica.

Geodesic Matting



Geodesic Matting

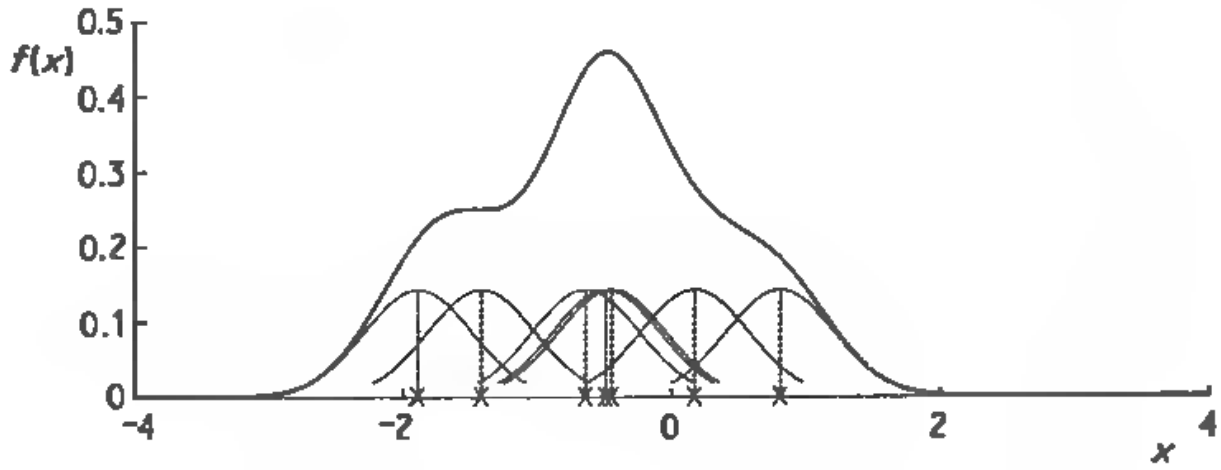


Figura: A curva maior mostra a f(x) estimada e as menores mostram os *kernels* individuais. Fonte: [Silverman 1986].

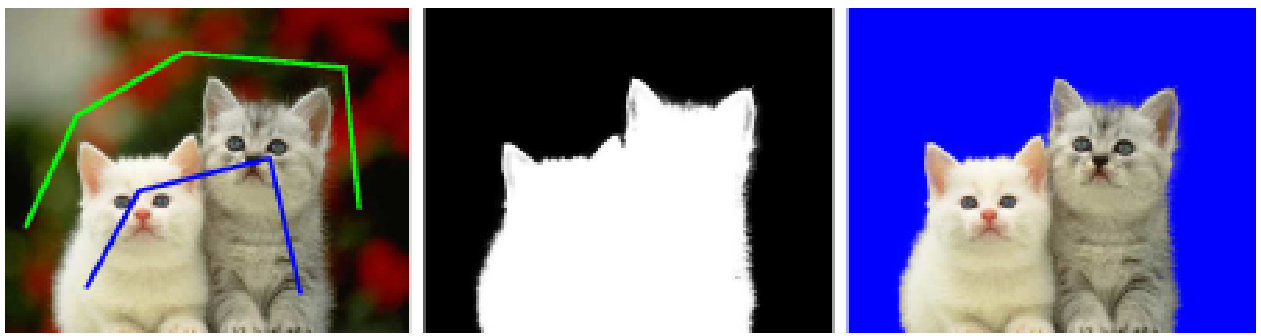
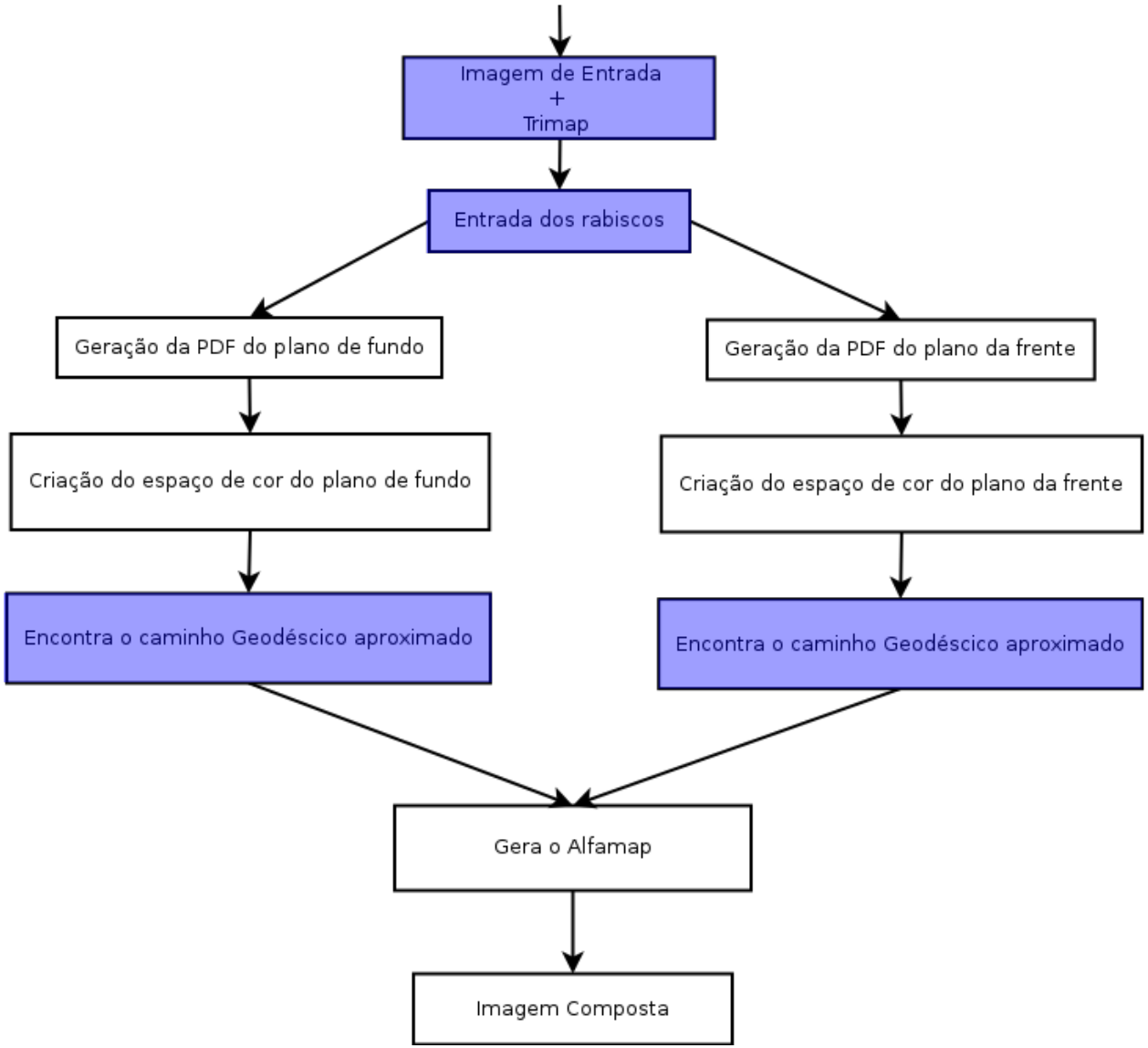


Figura: Exemplo do *Geodesic Matting*. Fonte: [Bai e Sapiro 2009].

Geodesic modificado



Geodesic modificado

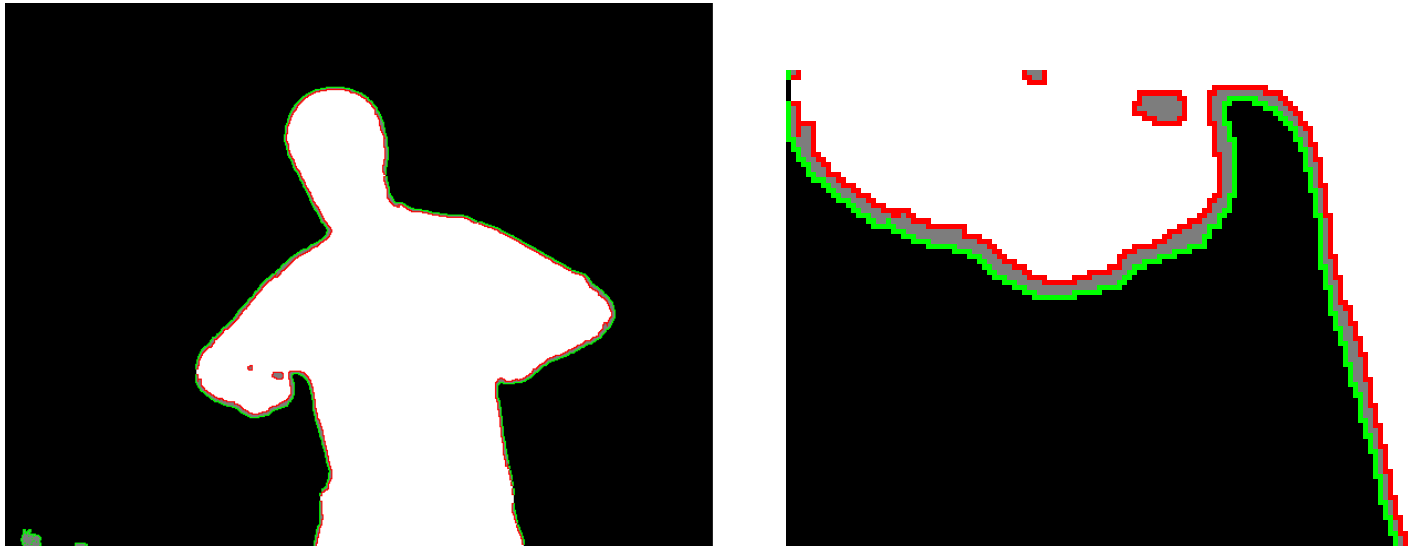


Figura: A cor verde representa os contornos do plano de fundo e a cor vermelha representa os contornos do plano da frente.

O menor caminho aproximado é encontrado por um algoritmo que caminha sempre para a posição com menor peso, sendo que o destino já é definido pela distância euclidiana.

APOD (*Assess*, *Parallelize*, *Optimize*, *Deploy*)

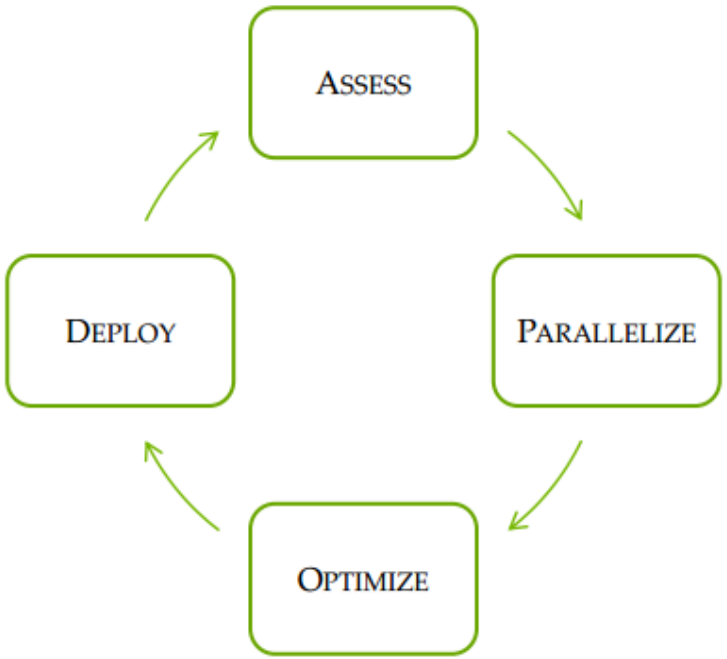


Figura: a técnica APOD, que é Avaliar, Paralelizar, Otimizar e Desenvolver. Fonte: [NVIDIA Corporation 2012]

Paralelização

Foram paralelizadas as seguintes partes:

- Algoritmo de *Chroma Key*;
- Geração automática de *trimap* utilizando o *Chroma Key*;
- Partes do algoritmo *Geodesic* Modificado:
  - Geração dos rabiscos;
  - Geração das PDFs;
  - Criação dos espaços de cor.

Paralelização

Primeiramente foi analisada a "capacidade de computação" das placas utilizadas e o tipo de processamento. A partir disso foi desenvolvido o código, melhorando assim o desempenho dos programas.

Paralelização

- Foi alocada memória na placa gráfica;
- Foram copiados os dados da memória do computador para a placa gráfica e da placa gráfica para a memória do computador;
- Para otimizar o programa, todos os *kernels* (funções em CUDA) foram executados com a ocupação máxima do multiprocessadores, por meio das configurações de *threads* por bloco e quantidade de blocos na grade;
- As otimizações devem respeitar as características da placa gráfica.

Imagem estática

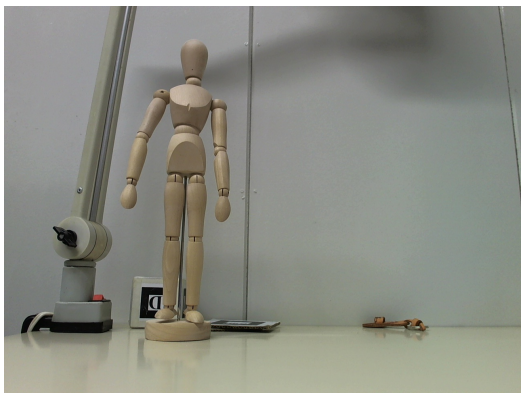
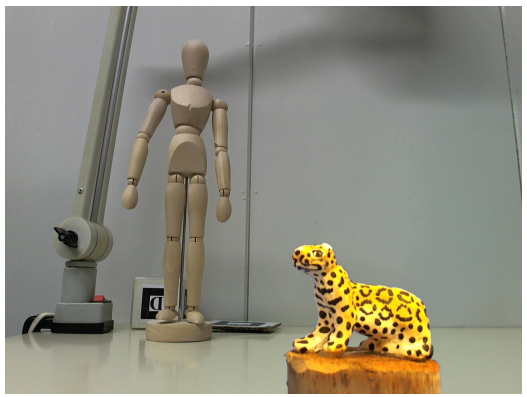




Tabela: Comparação de tempos utilizando o algoritmo de Van den Bergh & Lalioti (1999).

Resolução	CPU (ms)	GPU (ms)	GPU sem transferência de memória (ms)
2560x920	13,09	20,58	3,06
1280x960	2,98	5,50	0,79
640x480	0,78	1,46	0,24
320x240	0,16	0,45	0,09

Vídeo

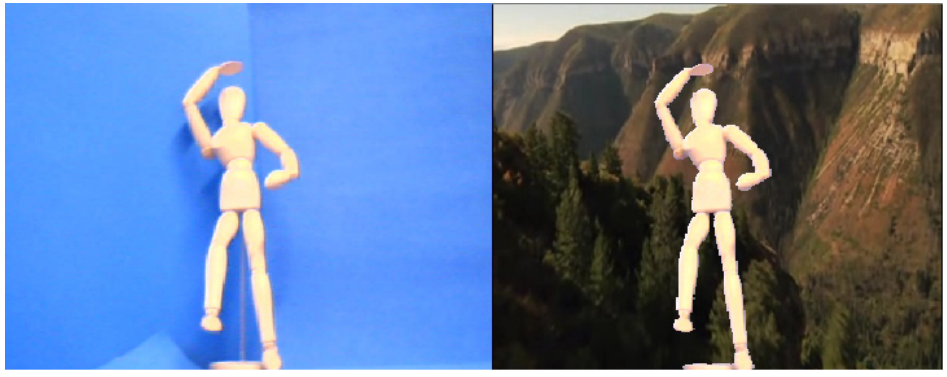


Tabela: Comparação de tempos em vídeos utilizando o algoritmo de Van den Bergh & Lalioti (1999).

Vídeo	Tamanho (largura, altura, quadros)	CPU (ms)	GPU (ms)	GPU sem transferência de memória (ms)
Vídeo1	160x120x353	0,099	0,278	0,023
Vídeo2	160x120x2191	0,099	0,280	0,024
Vídeo1	320x240x353	0,387	0,489	0,022
Vídeo2	320x240x2191	0,381	0,511	0,022
Vídeo1	640x480x353	1,429	1,441	0,033
Vídeo2	640x480x2191	1,456	1,460	0,035

Geração automática de *trimap*

Tabela: Comparação de tempos geração automática de *trimap*.

Resolução	CPU (ms)	GPU (ms)	GPU sem transferência de memória (ms)
2560x920	21,52	13,96	5,60
1280x960	5,45	3,72	1,40
640x480	1,1	0,99	0,38
320x240	0,47	0,31	0,13

Figura: Foto com plano de fundo azul.

Figura: Mapa de profundidade.

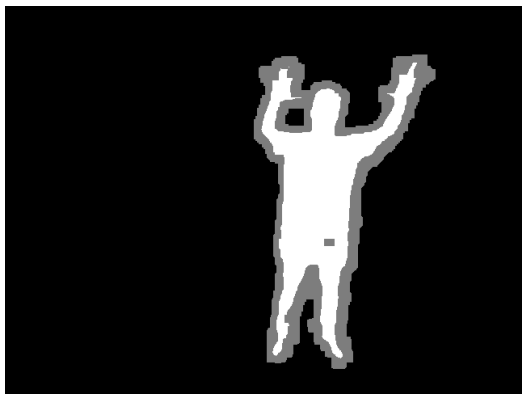


Figura: *Trimap* gerado pelo Kinect.

Figura: Recorte feito pelo *Baeysian Matting*.

Figura: Recorte feito pelo *Robust Matting*.

Segundo [Wang e Cohen 2007], o *Robust Matting* é considerado como o método de melhor desempenho (dentre os métodos analisados), entretanto o resultado mostra que com um *trimap* impreciso este pode ser pouco eficiente.

Figura: Foto original.

Figura: Trimap.

Figura: Recorte feito pelo *Baeysian Matting*.

Figura: Recorte feito pelo *Robust Matting*.

Figura: Recorte feito pelo *Geodesic* modificado.

Tabela: Comparação dos métodos com o *Robust Matting*.

Método utilizado	UIQI	SSIM
Bayesian Matting	0,9950	0,9997
Geodesic Matting modificado (implementação em CPU)	0,9993	0,9894
Geodesic Matting modificado (implementação em GPU)	0,9993	0,9894

Figura: Recorte feito pelo *Geodesic* modificado, com CPU.

Figura: Recorte feito pelo *Geodesic* modificado, com GPU.

Tabela: Comparação dos métodos.

UIQI	SSIM
0,9999	1,0000

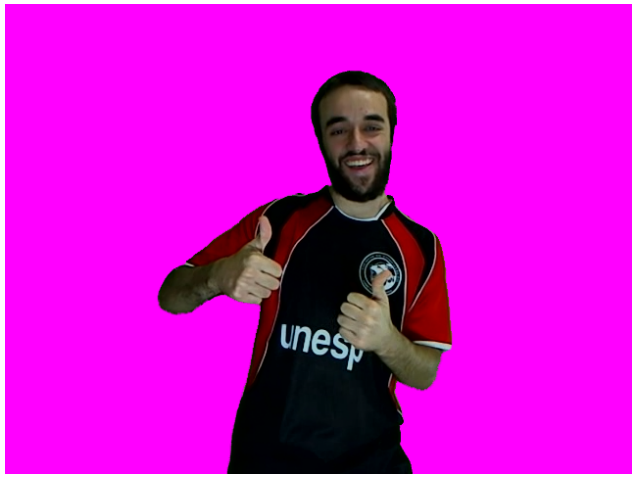


Figura: Figura de um quadro do vídeo.

Tabela: Implementação sequencial.

Tamanho (largura x altura)	FPS
320 x 240	42,4
640 x 480	13,4

Tabela: Implementação paralela.

Tamanho (largura x altura)	FPS
320 x 240	63,3
640 x 480	19,6

Conclusões

- O método proposto para o *pipeline* viabiliza a sua utilização em aplicações com restrições de tempo;
- A geração automática de *trimap* proposta neste trabalho pode ser utilizada em outros algoritmos já codificados em CUDA;
- As análises feitas mostram que o uso da arquitetura CUDA pode acelerar o algoritmo do *matting* digital, todavia a transferência de dados é o principal entrave;
- Os métodos propostos podem ser utilizados para produções de baixo custo;
- O Kinect não foi mais utilizado por ter uma qualidade de imagem muito inferior a da câmera, dessa forma optou-se por implementar o *pipeline* todo utilizando apenas esta.

Trabalhos Futuros

- Aplicação de outros métodos de resolução de área de incerteza (tais como, *Poisson Matting* e o *Random Walk*);
- Uso do *trimap* paralelizado juntamente com métodos já feitos para GPGPU, como o *Random Walk*, que segundo [Wang e Cohen 2007] já é implementado em GPGPU;
- Aperfeiçoar o algoritmo *Geodesic* modificado, visando maior desempenho com imagens de alta resolução, utilizando, para isto, a continuação da abordagem APOD no método paralelizado;
- Uso do Kinect como informação extra, acoplado juntamente com uma câmera fullHD, utilizando ou não a técnica de *chroma key*. Isto viabilizaria sua utilização em aplicações de Estúdios Virtuais e Videoconferência.

- Daniel Z. Vielmas, Antonio C. Sementille, Henrique F. de Arruda, Antonio C. T. Blaia Junior, João F. Marar; Geração automática de trimap utilizando chromakey Congresso de Iniciação Científica da Unesp. 2012.
- GULO, C. A. S. J.; Arruda, H.F.; SEMENTILLE, A. C. ;ARAUJO, A. F.; TAVARES, J. M. R.S. . Método de suavização de imagem baseado num modelo variacional paralelizado em arquitetura CUDA. In: GPU Computing Developer Forum, 2012, Curitiba, XXXII Congresso da Sociedade Brasileira de Computação. Porto Alegre: Sociedade Brasileira de computação, 2012.
- BLAIA JUNIOR, A.C. T.; VIELMAS, D.Z.; ARRUDA, H.F.; AGUILAR, I. A.; SEMENTILLE, A. C. . Automatic Trimap Generation with Structured Lighting and Chromakey. In SIBGRAPI 2012, Ouro Preto. Workshop of Works in progress, 2012.

- BAI, X.; SAPIRO, G. Geodesic Matting: A Framework for Fast Interactive Image and Video Segmentation and Matting. *International Journal of Computer Vision*, v. 82, n. 2, p. 113–132, 2009.
- BERGH, F. van den; LALIOTI, V. Software chroma keying in an immersive virtual environment. *South African Computer Journal*, Citeseer, v. 24, p. 155–162, 1999.
- CHUANG, Y. et al. A bayesian approach to digital matting. In: IEEE. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. [S.l.], 2001. v. 2, p. II–264.
- NVIDIA, C. Nvidia cuda c programming guide. *NVIDIA Corporation*, 2011.
- NVIDIA Corporation. *NVIDIA CUDA C Best Practices Guide*. 2012. Version 3.2.
- PORTER, T.; DUFF, T. Compositing digital images. *ACM Siggraph Computer Graphics*, ACM, v. 18, n. 3, p. 253–259, 1984.
- SILVERMAN, B. W. *Density estimation for statistics and data analysis*. London: Chapman and Hall, 1986.

- SMITH, A. R.; BLINN, J. F. Blue screen matting. In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1996. (SIGGRAPH '96), p. 259–268. ISBN 0-89791-746-4.
- SUN, J. et al. Flash matting. *ACM Trans. Graph.*, v. 25, n. 3, p. 772–778, 2006.
- WANG, J.; COHEN, M. Optimized color sampling for robust matting. In: IEEE. *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. [S.l.], 2007. p. 1–8.
- WANG, J.; COHEN, M. F. Image and Video Matting: A Survey. *Foundations and Trends in Computer Graphics and Vision*, v. 3, n. 2, p. 97–175, 2007.