

# Cálculo do Modelo de Fundo

Organização de Computadores – CC53B

Prof. Dr. Erikson Freitas de Moraes

**Resumo**—O Modelo de Fundo é uma das tarefas mais importantes quando se quer fazer a detecção de objetos ou analisar o movimento de objetos em uma cena. Muitas vezes é um processo custoso do ponto de vista do computador, apesar de ser uma tarefa simples para nós humanos. Existem algumas formas diferentes de calcular o modelo de fundo de uma cena. Neste trabalho o aluno irá implementar uma forma limitada e simples de calcular o modelo de fundo. Como será visto no texto a seguir, a proposta é usar um tipo de imagem fácil de ser lido ou escrito para que a implementação seja simplificada.

**Index Terms**—Organização de Computadores, Arquitetura de Computadores, Assembly MIPS-32, Vetores.



—



## 1 INTRODUÇÃO

Uma das tarefas mais usadas em processamento de imagens e visão computacional é o cálculo do modelo de fundo. Partindo de um modelo que represente o plano de fundo de uma cena, é possível detectar e separar facilmente objetos que sejam de interesse. Vários trabalhos na literatura utilizam a separação do modelo de fundo como tarefa inicial, como pode ser observado em [2], [6], [8], [1], [4], [7] e tantos outros.

Considerando uma câmera completamente estática, uma forma fácil de encontrar um modelo do plano de fundo é computando a média de alguns *frames*. De acordo com [6], o trabalho apresentado em [5], propôs usar como modelo de fundo a média dos últimos  $n$  *frames* de uma sequência, isso para cada pixel.

Dessa forma, o valor de cada pixel  $P_m(i, j)$  da imagem que representa o modelo de fundo é calculado por meio da média simples dos pixels de mesma coordenada nas  $n$  imagens da sequência escolhida, tal como apresenta a Equação 1:

$$P_m(i, j) = \frac{\sum_{k=0}^n P_k(i, j)}{n} \quad (1)$$

## 2 O PROBLEMA

A proposta deste trabalho é a implementação em assembly do MIPS-32 de um programa capaz de ler uma pequena sequência de arquivos do tipo PGM [3] e criar outro arquivo PGM como saída que contenha o modelo de fundo calculado com base no conjunto de entrada. A saída do programa deverá conter uma imagem do modelo de fundo da cena, tal como pode ser visto nas imagens de exemplo deste texto, onde se vê uma imagem com veículos e outra apenas contendo a cena sem os objetos que se movem.

## 3 O QUE DEVE SER FEITO

Neste trabalho o aluno deverá realizar pesquisa a respeito do tema, projetar e implementar um programa usando assembly do MIPS-32, capaz de ler um conjunto de arquivos PGM, processar e escrever outro arquivo PGM como saída. O arquivo de saída de conter uma imagem com o modelo de fundo da cena observada.

A cena escolhida pelo aluno pode ser gerada usando uma câmera RGB qualquer (a câmera do smartphone, por

- E. F. Moraes é Professor Dr. do Departamento Acadêmico de Informática, UTFPR, Ponta Grossa, PR.  
E-mail: emorais@utfpr.edu.br

exemplo ou outro disponível online). Como já foi mencionado, o modelo de fundo é calculado observando um pequeno conjunto de imagens que precisam constituir uma sequência temporal, ou seja, um vídeo. Depois de escolhido/gerado o vídeo, o aluno precisa separar os frames da sequência salvando cada um em um arquivo PGM separado. Para separar os frames, o aluno encontrará várias maneiras disponíveis online ou poderá usar softwares sem custo como virtualdub, irfanview ou avidemux, por exemplo. Vale ressaltar que algumas das alternativas vão precisar de algum plugin para lidar com os tipos de arquivos.

Depois de separados os frames, o aluno precisa desenvolver uma solução em Assembly do MIPS-32 para calcular uma imagem resultante que contenha um modelo de fundo para a cena escolhida.

O programa resultante deverá ser implementado usando o ambiente de simulação MARS<sup>1</sup> para criar um arquivo de extensão **asm** contendo o código do programa. O código resultante precisa contemplar os seguintes tópicos estudados em sala de aula:

- 1) acesso à memória;
- 2) operações aritméticas e lógicas;
- 3) operações para tomada de decisão;
- 4) chamadas ao sistema;
- 5) saltos condicionais e incondicionais e
- 6) suporte a funções no hardware.

#### 4 O QUE DEVE SER ENTREGUE

Como resultado do trabalho, o aluno deverá entregar dois arquivos através do sistema de submissão do **moodle**:

- 1) Um arquivo de extensão **asm** (nomeado como código.asm) contendo o código gerado pelo aluno, devidamente organizado e comentado para facilitar o perfeito entendimento durante o processo de avaliação;
- 2) Um arquivo de extensão **pdf** (nomeado como relatório.pdf) contendo um relatório<sup>2</sup> no formato de *short paper* de 2 páginas organizadas em 2 colunas e devidamente identificado, com nome, r.a. do aluno, título do trabalho e nome da disciplina. Os relatórios não identificados adequadamente serão desconsiderados. O relatório precisa conter as seguintes seções:
  - a) **Introdução:** Contextualização do problema e trabalhos relacionados. É muito importante que fale um pouco a respeito de fontes relacionadas ao tema;
  - b) **Desenvolvimento:** Métodos e implementações utilizados durante o trabalho. Explicar como o problema foi resolvido;
  - c) **Resultados:** Mostrar exemplos de execução como resultado do trabalho. Esta seção deve mostrar que o programa gerado realmente funciona;

1. O Programa *Mars Simulator* que pode ser encontrado no seguinte endereço:  
<http://courses.missouristate.edu/KenVollmar/mars/>

2. Muito parecido com este texto e seguindo modelo indicado pelo professor!

- d) **Referências:** Lista de referências **bibliográficas** utilizadas durante o trabalho. **Não** use link para vídeos ou blogs.

Os dois arquivos resultantes do trabalho devem ser compactados em um arquivo único do tipo **zip** nomeado com o r.a. do aluno (por exemplo, para o r.a. 000000 **000000.zip**). Este arquivo final deverá ser submetido no sistema de submissão do moodle até a data agendada no sistema.

É importante atentar para a possibilidade de haver, total ou parcialmente, correção automática do trabalho. Por isso, é extremamente necessário o cumprimento dos protocolos de entrega, tal como nomes de arquivos, formatos e identificação.

#### 5 PONTUAÇÃO

Ao trabalho será atribuída uma nota no intervalo [0..10], que representará **20%** da nota final da disciplina. A nota será computada considerando o texto produzido, corretude do resultado do programa, uso correto da arquitetura estudada e atendimento aos requisitos apontados neste documento.

#### 6 PONTOS EXTRAS

Serão agraciados com **até 2 pontos extras** os trabalhos que, além do exposto na proposta deste documento, desenvolvem o mesmo mecanismo para arquivos do tipo PPM.

Nesse caso, o aluno deverá incluir uma nova seção no relatório final chamada **Ponto extra**, onde devem ser apresentadas as considerações e resultados adequados [3].

#### REFERÊNCIAS

- [1] BARNICH, O., AND VAN DROOGENBROECK, M. Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image processing* 20, 6 (2011), 1709–1724.
- [2] ELGAMMAL, A., HARWOOD, D., AND DAVIS, L. Non-parametric model for background subtraction. In *European conference on computer vision* (2000), Springer, pp. 751–767.
- [3] HENDERSON, B. Netpbm home page. *Electronic document at* <http://netpbm.sourceforge.net/>, edit date 8, 01 (2008).
- [4] LEE, D.-S. Effective gaussian mixture learning for video background subtraction. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 5 (2005), 827–832.
- [5] LO, B., AND VELASTIN, S. Automatic congestion detection system for underground platforms. In *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing. ISIMP 2001 (IEEE Cat. No. 01EX489)* (2001), IEEE, pp. 158–161.
- [6] PICCARDI, M. Background subtraction techniques: a review. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)* (2004), vol. 4, IEEE, pp. 3099–3104.
- [7] SEN-CHING, S. C., AND KAMATH, C. Robust techniques for background subtraction in urban traffic video. In *Visual Communications and Image Processing 2004* (2004), vol. 5308, International Society for Optics and Photonics, pp. 881–893.
- [8] ŽIVKOVIC, Z. Improved adaptive gaussian mixture model for background subtraction. In *null* (2004), IEEE, pp. 28–31.