

Detecção de veículos em vídeos de trânsito

Gabriel Coelho da Silva

Departamento de Engenharia Eletrônica

Faculdade Gama - UnB

Brasília, Brasil

Email: csilvagabriel@hotmail.com

Julia Borges Silva

Departamento de Engenharia Eletrônica

Faculdade Gama - UnB

Brasília, Brasil

Email: 170014355@aluno.unb.br

Abstract—The abstract goes here.

I. INTRODUÇÃO

É estimado que cerca de 40% da população passa, pelo menos, uma hora do seu dia em trânsito [?]. Como uma das formas de otimizar os sistemas de transporte, inovações baseadas em Tecnologias da Informação e Comunicação (TIC) vem sendo utilizadas como propulsores de inovação. Desta forma, temos que a aplicação de um conjunto de soluções TIC relacionado aos modais de transportes como os Sistemas de Transportes Inteligentes (ITS - *Intelligent Transport System*) [?].

As aplicações ITS podem ser subdivididas em algumas categorias, a depender do seu objetivo, como obter informações dos viajantes, gerenciar o tráfego, realizar controle de preços e pagamentos, entre outros [?]. Além disto, podem ser segmentados de acordo com o tipo de dado em que é feito o processamento, como dados visuais (vídeos e imagens) [?], que será o foco do nosso trabalho.

Aprofundando o foco em informações visuais, diversas são as aplicações desenvolvidas, como detecção, acompanhamento e reconhecimento de objetos de tráfego, análise de comportamento de tráfego, estimação de densidade de veículos e pedestres, construção de trajetória veicular e análise estatística de dados de tráfego [?].

Além disto, com o advento da computação em nuvem e do Big Data, a tendência é que haja uma transição entre a utilização de um hardware poderoso e caro para diversos sistemas computacionais mais simples, denominados como nós [?]. Dito isto, a proposta deste trabalho é a de implementar um sistema operacional embarcado capaz de realizar a detecção de veículos em movimento, bem como a estimação de densidade de veículos em um dado espaço de tempo, a partir de observações obtidas de câmeras de monitoramento de trânsito.

II. DESENVOLVIMENTO

A. Descrição de hardware

Para embarcar o sistema operacional, será utilizada da Raspberry Pi 3B+, um *Single-Board Computer* desenvolvido pela Raspberry Pi Foundation em associação com a Broadcom. Com um processador de quatro núcleos e 64-bits rodando a 1.4Ghz, duas bandas de W-LAN, memória de 1GB SDRAM e conexões GPIO [?], este computador em placa única é capaz de realizar processamentos mais pesados que os sistemas microcontrolados, enquanto é uma solução mais viável do que

um microcomputador de mesa. Suas especificações técnicas são [?]:

- **Processador:** Broadcom BCM2837B0, Cortex A-53 64-bit SoC @ 1,4GHz;
- **Memória:** 1GB LPDDR2 SDRAM;
- **Conectividade:** 2.4 GHz e 5GHz IEEE 802.11.b.g.n.ac LAN sem fio, Bluetooth 4.2, BLE;
- **Acesso:** 40 pinos GPIO;
- **Multimídia:** H.264, decoder MPEG-4 (1080p30), encoder H.263 (1080p30), OpenGL ES 1.1, 2.0 graphics.

B. Descrição de software

Para a detecção de veículos em movimento, utiliza-se *background subtraction*, que consiste na diferença entre o frame atual e um modelo de *background*. Após isso será criada uma máscara binária, em que os contornos dos objetos serão detectados e, após a definição de uma área de saída de automóveis, contagem dos veículos passando na filmagem de uma via. Já para a contagem dos veículos **COMPLEMEN-**
TAR

C. Modelo inicial de background

Inicialmente, uma imagem de *background* é necessária para comparação inicial e atualização do modelo de *background*. O cálculo dos valores do pixel é feito a partir da mediana entre pixels de mesma posição em uma quantidade n de *frames* do vídeo.

D. Imagem de diferença e máscara binária

Utilizando a técnica de subtração de *background*, a variação dos valores dos pixels ao longo do vídeo é feita. Essa variação, em geral, ocorre devido ao movimento de objetos, porém, pode também ser causada pela variação de iluminação, vento balançando a vegetação próxima à rodovia, etc.

Essa imagem de diferença criada passa por um processo de limiarização com o método de Otsu para criação de uma imagem binária para segmentação dos veículos.

Para refinar a máscara e garantir que toda a área dos veículos serão detectadas, são utilizadas operações morfológicas, sendo elas: abertura, dilatação, fechamento e preenchimento. A abertura é responsável por retirar ruídos indesejados, causados pela mudança de iluminação e movimento de vegetação. A dilatação liga áreas próximas para que

todo o veículo seja detectado no mesmo *blob* da máscara; já o fechamento e preenchimentos conectam pequenos espaços abertos para refinar a máscara.

E. Atualização de background

Para a atualização, um modelo instantâneo (Ib_t) é criado utilizando a máscara do t -ésimo *frame*. Um *pixel* na posição $[i][j]$ do *frame* atual (Ci_t) é amostrado quando a máscara (Dt) tem valor zero, ou seja, é um pixel preto, e o pixel do *background* atual (Cb_t) é amostrado quando a máscara tem valor 1 Ou seja, uma imagem de fundo é criada amostrando a imagem atual e substituindo as áreas com automóveis por pixels de fundo do modelo. O processo pode ser ilustrado a partir da equação:

$$Ib_t[i][j] = \begin{cases} Ci_t[i][j], & \text{se } Dt[i][j] = 0 \\ Cb_t[i][j], & \text{se } Dt[i][j] = 1 \end{cases} \quad (1)$$

Por último, o modelo atualizado (Cb_{t+1}) é gerado utilizando uma soma ponderada do background instantâneo e o modelo anterior, ou seja,

$$Cb_{t+1}[i][j] = \alpha Ib_t[i][j] + (1 - \alpha)Cb_t[i][j] \quad (2)$$

O fator multiplicativo α afeta a velocidade de atualização do modelo, que deve ser rápido o suficiente para captar mudanças de iluminação e lento o bastante para que mudanças momentâneas não afetem as imagens subsequentes. O valor $\alpha = 0.01$ foi escolhido empiricamente considerando o *trade-off* anterior.

III. CONCLUSÃO

The conclusion goes here. Conclui-se que o essa matéria é um inferno :)