

UNIVERSIDADE DE SÃO PAULO  
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO

**Projeto Final**  
**Identificação de Movimento em Imagens Através de  
Fluxo Ótico**

*Processamento de Imagens - SCC5830*

Prof: Moacir Ponti

Laíse Aquino Silva - 7986924

São Carlos  
2018

## Introdução

Este projeto consiste na implementação em Python de um método para identificar o movimento de elementos em sequências de imagens consecutivas, sendo possível reconhecer a direção em que os mesmos estão se movendo em uma determinada cena. As imagens utilizadas foram retiradas do conjunto *UCSD Anomaly Detection Dataset - Peds2*<sup>1</sup>, que mostram uma sequência de fotos tiradas de uma rua na qual há um fluxo de pessoas predominantemente no sentido direita-esquerda, como visto na Figura 1.

O objetivo do projeto é que o algoritmo desenvolvido possa futuramente ser otimizado para aplicação em sistemas embarcados que possuam várias unidades de processamento, a exemplo de câmeras de segurança ou drones baseados em arquiteturas heterogêneas (FPGAs, CPUs, etc).



Figura 1 – Imagem da sequência original

## Metodologia

O movimento da sequência de imagens foi representado como vetores de movimento calculados entre *frames* consecutivos por um método de Fluxo Ótico, sendo utilizado especificamente o algoritmo de Lucas-Kanade<sup>2</sup>.

As imagens originais, que consistiam em arquivos *.tif* em escala de cinza e dimensão 240x360, foram pré-processadas para remover possíveis ruídos e variações indesejadas tanto nas imagens individuais quanto na sequência como um todo. Para isso, cada imagem foi carregada como um *numpy array* e passou por um filtro Gaussiano de janela 5 e parâmetro *sigma* 1, sendo em seguida normalizadas e armazenadas em uma lista de *frames*.

O conjunto de *frames* sofreu uma filtragem temporal através de um filtro de mediana, que considera uma janela de 5 imagens para calcular a imagem resultante na sequência filtrada. Em ambos os casos, foi aplicado *padding* de bordas dos elementos a serem filtrados, isto é, repetindo os pixels das bordas das imagens para o

---

<sup>1</sup> <http://svcl.ucsd.edu/projects/anomaly/dataset.htm>

<sup>2</sup> <https://www.datasciencecentral.com/>

filtro Gaussiano e repetindo os primeiros e últimos *frames* da sequência para o filtro de mediana. Por fim, as imagens resultantes são armazenadas em formato *.png*.

Após o pré-processamento, a sequência de imagens é carregada *frame a frame* de maneira análoga à captura de imagens por uma câmera em tempo real. O método de Lucas-Kanade foi aplicado em cada dupla de imagens consecutivas para obter seu fluxo ótico, que foram armazenados como vetores correspondentes a cada *pixel*. Esses vetores são exibidos na tela acima da imagem sendo processada, de forma menos densa para facilitar sua visualização, acompanhados de sua representação em cores em outra janela, calculada com base nos ângulos aos quais estes vetores correspondem na roda de cores HSV. Os passos de carregamento e visualização foram realizados com o auxílio de métodos da biblioteca OpenCV.

## Resultados

Sem um pré-processamento adequado, os resultados obtidos inicialmente não apresentaram o comportamento esperado pelo método, que produziu fluxos inconsistentes, como visto nas Figuras 2 e 3. Consequentemente, foram geradas imagens onde não era possível identificar claramente o movimento de pessoas.

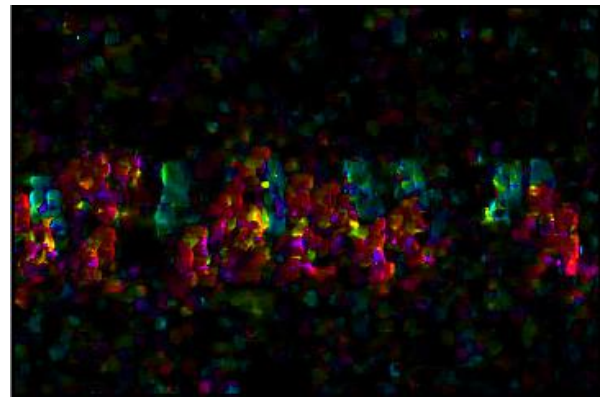


Figura 2 – Melhor caso inicial

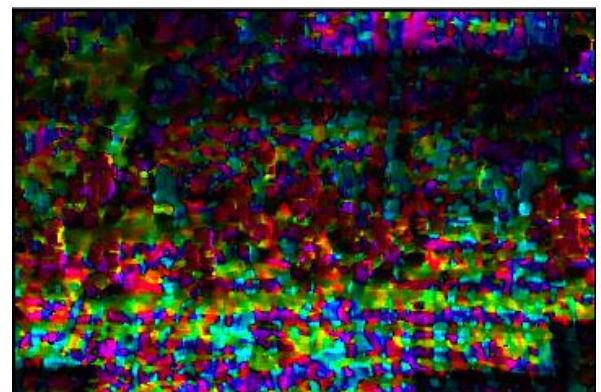


Figura 3 – Caso comum inicial

Após aplicação de filtragem temporal na etapa de pré-processamento, o melhor caso tornou-se bem mais comum, apresentando formas mais nítidas e cores que permitem diferenciar o movimento dos elementos na imagem, como mostra a Figura 4.

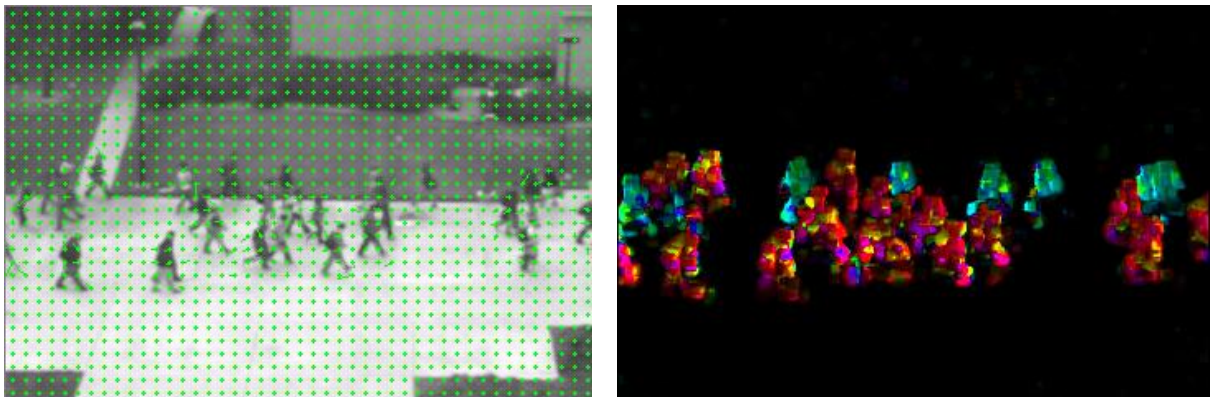


Figura 4 – Fluxos melhor definidos após filtragem temporal

Entretanto, o tempo necessário para calcular o fluxo ótico de cada *frame* através do método descrito mostrou-se muito lento para ser utilizado em uma aplicação envolvendo a coleta de imagens em tempo real. Além disso, o resultado ainda será caótico até serem processados os primeiros *frames* que não sofreram filtragem temporal completa devido aos índices da janela do filtro.

Quando comparado a um método semelhante – o algoritmo de Gunnar-Farneback já implementado na biblioteca OpenCV – observa-se que é possível atingir tempos de resposta mais adequados para aplicações em tempo real ao sacrificar precisão, mas ainda obtendo uma representação geral do movimento dos elementos de uma cena como mostra a Figura 5. Logo, possíveis extensões deste projeto seriam explorar as opções de paralelismo e generalização do método de Lucas-Kanade implementado, observando mais detalhadamente seu desempenho frente a outros métodos de estimação de fluxo ótico.

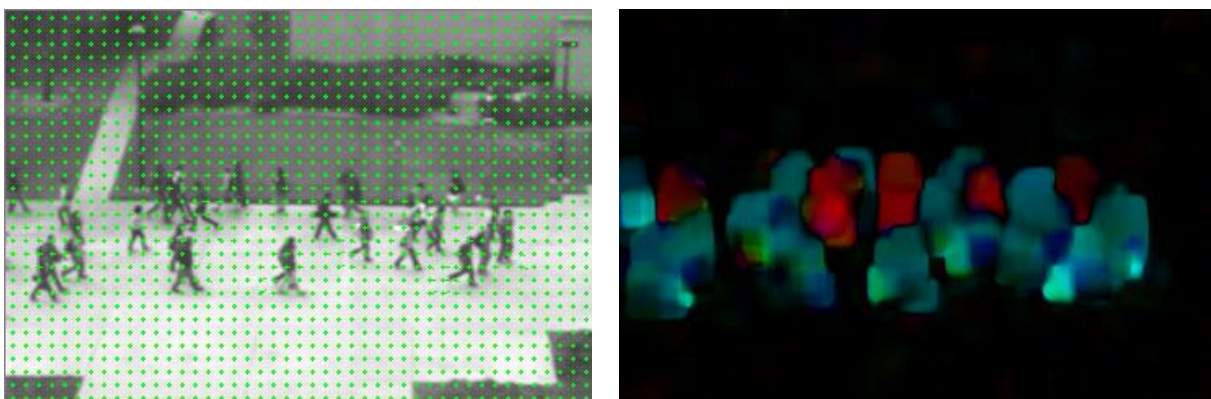


Figura 5 – Método de Gunnar-Farneback resulta em fluxos menos detalhados de forma mais rápida