INSTITUTO NACIONAL DE MATEMÁTICA PURA E APLICADA

HALLISON OLIVEIRA DA PAZ

CURSO DE PROCESSAMENTO DE IMAGENS

Assignment 4: Tracking

RIO DE JANEIRO

2015

1 Introdução

Este relatório descreve a implementação do *assignment Tracking* do curso de Processamento de Imagens. Foram utilizados a biblioteca OpenCV (3.0) para implementação das funções de visão computacional e o ambiente de desenvolvimento integrado QT Creator para a confecção das interfaces gráficas. Em particular, para o cálculo do *Optical Flow* denso foi utilizado o algoritmo de *Farneback* [4]. O código fonte encontra-se no repositório [1].

2 Explicação do Algoritmo

O Algoritmo de Farneback é um algoritmo para estimação de optical flow denso, ou seja, o algoritmo tenta determinar uma estimativa de movimento para todos os pixels da imagem. Uma abordagem um pouco diferente seria o cálculo de optical flow esparso, seja por meio da detecção de features dentro das imagens e o acompanhamento do movimento dessas features, seja por meio de uma suposição de continuidade de movimento de regiões próximas, na qual realiza-se o cálculo do optical flow por blocos dentro da imagem.

De posse de uma sequência ordenada de imagens e sob algumas hipóteses podemos realizar uma estimativa de movimentos dentro da cena. Supondo que a primeira imagem seja capturada em um tempo t e a segunda em $t + \triangle t$, considerando que há uma "constância de brilho" entre as duas imagens podemos calcular uma estimativa de movimento aparente baseada em uma aproximação local polinomial do sinal imagem.

Embora a ideia seja bem simples e funcione razoavelmente bem dentro de um ambiente controlado, é importante notar que muitas variáveis podem causar dificuldades nesta estimação, dentre elas:

- Falha da hipótese de constância de brilho: é possível que haja uma mudança significativa de iluminação entre um frame e outro, principalmente se a cena envolver objetos e fontes de iluminação que favoreçam o surgimento de brilho especular.
- Movimentos de grande amplitude: é possível que alguns objetos estejam se locomovendo em alta velocidade dentro da cena e, mesmo em um intervalo pequeno de tempo, seu deslocamento gere uma grande variação entre um frame e outro.
- Oclusão: evidentemente, não é possível obter informações de movimento de elementos que não estavam em uma das imagens por conta de oclusão ou por estarem fora do campo de visão de uma delas. É preciso levar isto em consideração para aumentar a robustez dos algoritmos.

Por conta dessas dificuldades, vários algoritmos de estimação de movimento baseados em cálculo de optical flow utilizam-se de alguns artifícios adicionais para aumentar sua eficácia e robustez. Por exemplo, o algoritmo de Farneback realiza uma expansão em níveis de pirâmide multiresolução para tentar detectar movimentos de maior amplitude. O número de níveis de pirâmide, incluindo a imagem original, é um dos parâmetros do algoritmo. O tamanho da janela de busca também é um dos parâmetros do algoritmo. Quanto maior a janela de busca, maior a chance de detectar movimentos de grande amplitude (e maior o esforço computacional também, pois a janela é tomada para cada pixel).

O algoritmo de Farneback adota uma modelagem inicial bastante simplificada, em que cada imagem é aproximada por um mesmo polinômio quadrático que difere apenas por um deslocamento. Esse modelo, contudo, pode ser refinado por um conhecimento a priori do tipo de movimento da cena, que pode ser dado como entrada para o algoritmo. Dessa forma, é possível uma abordagem de malha fechada em um processo iterativo que, ao calcular uma estimativa de optical flow num dado passo, insere a última

estimativa calculada como entrada para uma próxima iteração. O número de iterações executadas é outro parâmetro do algoritmo.

O tamanho da janela de pixels utilizada para calcular uma expansão polinomial local do sinal imagem também é um dos parâmetros do algoritmo *(poly_ n).* Maiores valores de poly_n resultam em uma aproximação da imagem por superfícies mais suaves, gerando mais robustez ao custo de maior borramento no campo de movimento. Durante o cálculo da expansão polinomial, as derivadas são suavizadas por uma função gaussiana cujo desvio padrão é especificado pelo parâmetro poly_sigma. Em geral, para poly_n=5, poly_sigma em torno de 1,1 é satisfatório; para poly_n=7, ajusta-se poly_sigma para 1.5.

Estes parâmetros estão expostos na barra inferior da interface do programa. O parâmetro que representa a razão entre as imagens no cálculo multiescala não foi exposto, de modo que assume-se sempre a criação de uma pirâmide clássica em que as dimensões da imagem são divididas por dois a cada nível que se incrementa.

3 Implementação

O programa implementado permite que o usuário selecione duas imagem em seu computador ou utilize a câmera principal do sistema para capturar o par de imagens.

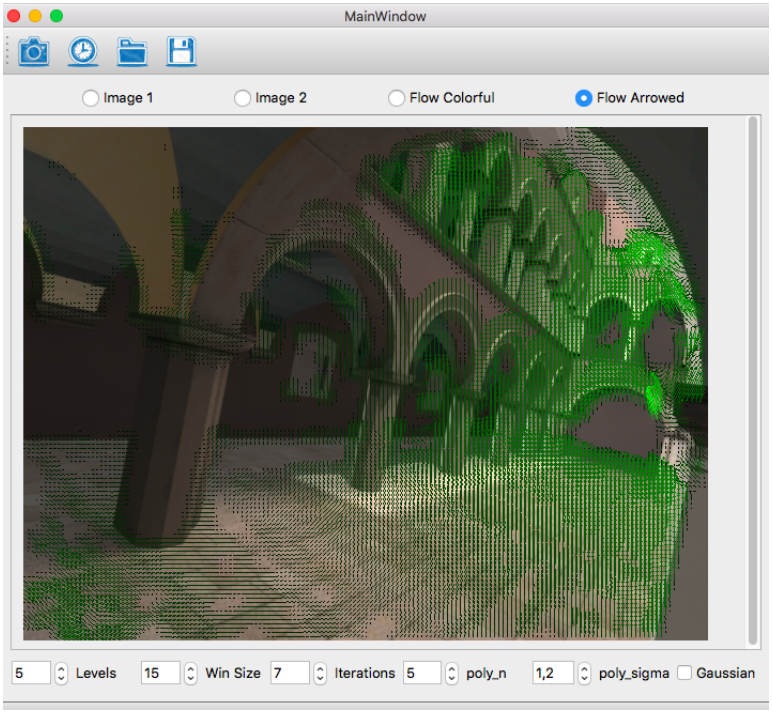


Figure 1: Interface do programa

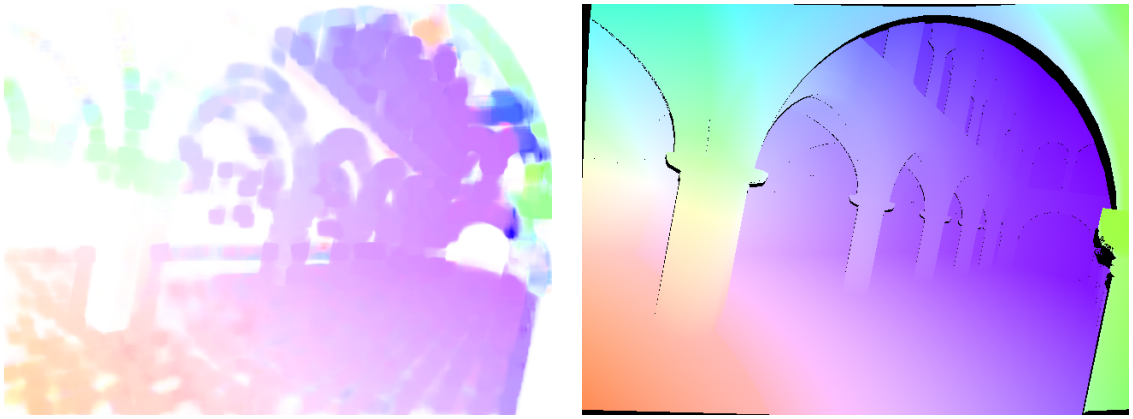
O usuário deve selecionar se deseja carregar a imagem 1 ou 2 e, então, clicar no ícone de "pasta" na barra superior da interface. Dessa forma, será exibida uma janela para seleção de arquivo, onde pode-se escolher qual imagem será aberta e exibida na área principal do programa. Para capturar uma imagem a partir da câmera, deve-se clicar no ícone da câmera para ativá-la e, quando encontrar a posição desejada da câmera, deve-se clicar novamente no ícone para desativá-la. O último frame capturado pela câmera no momento da desativação será a imagem utilizada.

Após selecionar as duas imagens, o usuário pode clicar em uma das opções Flow Colorful ou Flow Arrowed para exibir um mapa colorizado do optical flow ou um mapa de vetores, respectivamente.

Os *radio buttons* da interface permitem que o usuário retorne a qualquer uma das imagens já carregada a qualquer momento.

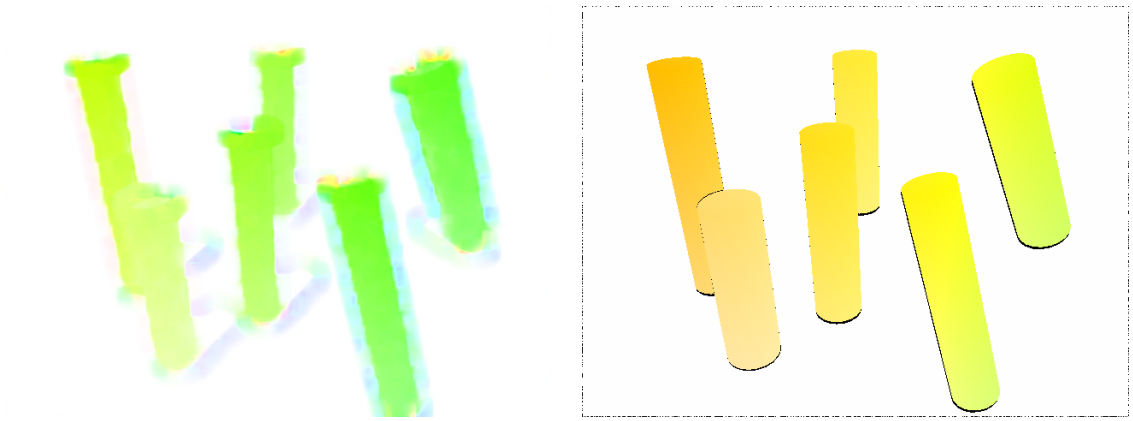
Os mapas de optical flow apenas são recalculados se houver alguma mudança de parâmetros. Portanto, caso o usuário esteja visualizando o mapa de cor de optical flow e deseje visualizar o mapa de vetores, é necessário que ele altere algum parâmetro na parte inferior da interface (mesmo que ele retorne o valor para o original) para que o programa entenda que é necessário recalcular a imagem. Isto foi realizado desta forma, pois pode ser muito custoso recalcular o optical flow denso de um par de imagens; então, caso o usuário queira visualizar as imagens originais e um dos mapas de optical flow intervaladamente, não haverá necessidade de recalculá-lo.

A título de comparação, utilizou-se algumas imagens do *UCL Ground Truth Optical Flow Dataset v1.2 [3],* que constitui-se de imagens sintéticas que servem como "ground truth" para validação de algoritmos para cálculo de optical flow. As figuras 2, 3 e 4 mostram o resultado para algumas comparações.



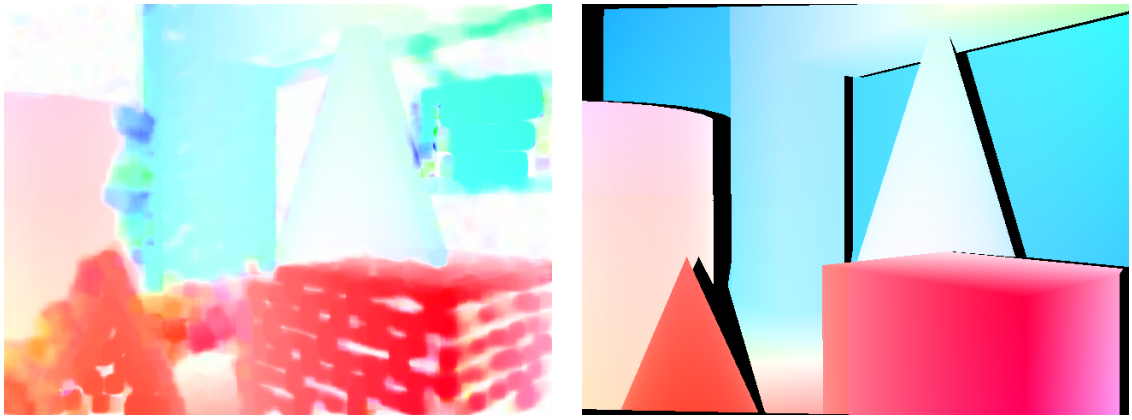
(a) Resultado neste trabalho (b) Ground Truth

Figure 2: Comparação entre o resultado obtido no trabalho e o dataset sintético (018_Sponza1)



(a) Resultado neste trabalho (b) Ground Truth

Figure 3: Comparação entre o resultado obtido no trabalho e o dataset sintético (051_blow1Txtr1)



(a) Resultado neste trabalho (b) Ground Truth

Figure 4: Comparação entre o resultado obtido no trabalho e o dataset sintético (026_Brickbox1t1)

References

[1] Hallison's github: [internet] Disponíel em: <https://github.com/hallpaz/Image-Processing-IMPA-2015>

[2] How to draw Optical flow images from ocl::PyrLKOpticalFlow::dense(). [internet] Acessado em 14 de dezembro de 2015. Disponível em: <http://stackoverflow.com/questions/20064818/how-to-draw-optical-flow-images-from-oclpyrlkopticalflowdense>

[3] UCL Ground Truth Optical Flow Dataset v1.2. [internet] Acessado em 13 de dezembro de 2015 http://visual.cs.ucl.ac.uk/pubs/flowConfidence/supp/

[4] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In Image Analysis, pages 363–370. Springer, 2003.