

# Kanade-Lucas-Tomasi Tracker

---

Ricardo Marroquim

**data entrega:**

pós-graduação: 18/05/2016

graduação: 27/05/2016

## 1 NOTAÇÕES

Uma imagem é definida como uma função  $f(x, y), x, y \in \mathbb{Z}$ , onde  $f(x, y) : \mathbb{Z}^2 \rightarrow \mathbb{R}$ . A função retorna um escalar no intervalo  $0 \leq f(x, y) \leq 1$ .

Em outras palavras, consideramos que a imagem é um espaço discreto e pode ser acessada em qualquer posição inteira, e o valor de intensidade de cada pixel está no intervalo entre zero e um.

Neste texto iremos omitir muitas vezes os parâmetros para facilitar a leitura:  $f = f(x, y)$ .

Alguns cuidados:

- dependendo da biblioteca que você está usando, o valor de um pixel deverá ser definido no intervalo inteiro  $[0, 255]$ , neste caso deve ser feita a devida conversão
- atenção com o tipo da estrutura de dados, algumas bibliotecas criam a imagem como uma matriz de bytes (ex. *unsigned char*), outras podem criar como *float* ou *int*

## 2 OBJETIVO

Implementar um algoritmo de Tracking usando fluxo ótico de Lucas-Kanade. A entrada do seu programa deve ser uma série de quadros de um vídeo. Você pode ler diretamente o vídeo

se preferir. No primeiro quadro deve ser selecionada uma região da imagem para realizar o tracking.

A saída do seu programa deve ser a série de quadros com a região anotada (um retângulo desenhado em cada uma).

Os artigos originais estão listados no final do documento, mas existem muitas fontes na internet que podem ajudar.

Existem algumas sub-tarefas que você deve seguir antes de tentar realizar o tracker completo:

- Fluxo Ótico Lucas-Kanade simples para dois quadros
- Fluxo Ótico Lucas-Kanade em pirâmide para dois quadros
- KLT tracker em pirâmide para região em múltiplos quadros

### 3 FLUXO ÓTICO LUCAS-KANADE

#### 3.1 PONTOS PARA FAZER O TRACKING

A formulação de fluxo ótico de Lucas-Kanade assume intensidade constante para o mesmo pixel em quadros diferentes. A aproximação de primeira ordem usando série de Taylor do movimento de um pixel durante um tempo  $\Delta t$  pode ser escrito da seguinte forma:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t$$

Assumindo que a intensidade do pixel não muda entre os dois quadros:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t)$$

Temos que:

$$\frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t = 0$$

Vamos reescrever como:

$$(I_x, I_y) \cdot (v_x, v_y) = -I_t$$

onde queremos encontrar o fluxo  $(v_x, v_y)$ . Como vimos, não conseguimos resolver diretamente pois temos uma equação e duas incógnitas. Portanto, usamos os pixels vizinhos e montamos um sistema linear  $Av = b$ .

$$\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = - \begin{bmatrix} I_{t1} \\ I_{t2} \\ \vdots \end{bmatrix}$$

E podemos resolver da seguinte forma:

$$\vec{v} = (A^T A)^{-1} A^T b$$

Note que  $A^T A$  é uma matriz  $2 \times 2$ , e  $A^T b$  é uma matrix  $2 \times 1$ :

$$(A^T A)_{2 \times 2} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

$$(A^T b)_{2 \times 1} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

Para melhor a estimativa você deve usar pesos nas somas acima, para que os pixels mais próximos do centro tenham mais influência no cálculo do fluxo. Use por exemplo pesos Gaussianos normalizados (para que a soma dos pesos  $\approx 1$ ). Você notou diferença com ou sem pesos?

Para os pontos onde  $A^T A$  for inversível e bem-condicionada, será possível encontrar o valor do fluxo. Para ter uma boa seleção dos pontos a serem seguidos pelo tracker, siga os seguintes passos:

1. Compute a matriz  $A^T A$  e o seu autovalor mínimo  $\lambda_m$  para cada ponto da imagem
2. Encontre  $\lambda_{max}$  entre todos os  $\lambda_m$
3. Selecione os pontos que possuïrem  $\lambda_m > 0.1 * \lambda_{max}$
4. Destes pontos acima, elimine pontos muito próximos (um ponto é mantido somente se ele for máximo em  $\lambda_m$  em uma região  $3 \times 3$ )
5. opcional: garante uma distância mínima entre os pontos escolhidos (ex. 5 ou 10 pixels)

#### Dicas:

- Utilize gradientes simples (não precisa ser Sobel):

$$I_x(i, j, 0) = 0.5 * (I(i + 1, j) - I(i - 1, j))$$

$$I_y(i, j, 0) = 0.5 * (I(i, j + 1) - I(i, j - 1))$$

- Para o gradiente temporal, subtraia uma image da seguinte

$$I_t^0(i, j) = I^1(i, j) - I^0(i, j)$$

- Não precisa fazer tratamento das bordas (o fluxo nas bordas geralmente não é preciso)

### 3.2 ENCONTRANDO O FLUXO

Para os pontos escolhidos na seção anterior, uma vez invertida a matriz o fluxo pode ser encontrado diretamente:

$$\vec{v} = (A^T A)^{-1} A^T b$$

Você pode encontrar alguns datasets de teste (com 8 quadros) no seguinte link: <http://vision.middlebury.edu/flow/data/>

Alguns pares de imagens testes mais simples também podem ser encontradas aqui: <http://demo.ipol.im/demo/20/>

Note que esses datasets são para fluxos densos, e o Lucas Kanade é um fluxo esparsos. De qualquer forma será útil para testar o seu algoritmo.

### 3.3 ALGORITMO EM PIRÂMIDE

Uma vez que o fluxo ótico em um nível esteja pronto, você deve utilizar uma implementação em pirâmide para permitir o cálculo de deslocamentos maiores. Para tanto, você pode usar a sua implementação de pirâmide Gaussiana do último trabalho.

Três pirâmides são necessárias, uma para primeira imagem, uma para a segunda imagem, e uma para o fluxo (que será preenchida de cima para baixo);

O algoritmo é essencialmente o mesmo. Para os pontos escolhidos na Seção 3.1, comece computando o fluxo no nível de menor resolução (nível mais alto da pirâmide). Note que um ponto na posição  $(i, j)$ , estará na posição  $i/2^L, j/2^L$  no nível  $L$ .

A única mudança necessária é: inicializar o fluxo de cada nível com a estimativa do fluxo no nível superior. Lembrando que, se  $v^L$  é o fluxo no nível  $L$ , então  $v^{L-1} = 2 * v^L$ .

### 3.4 REALIZANDO O TRACKING

O tracking será realizado "seguindo" os pontos encontrados na Seção 3.1 durante os demais quadros. Durante a execução alguns pontos serão descartados (ex. pontos que saíram dos limites da imagem) e outros incluídos.

Esses procedimentos serão descritos aqui em muito breve ...