Lucas-Kanade(LK) 光流法

Dezeming Family

2023 年 3 月 14 日

DezemingFamily 系列文章和电子书**全部都有免费公开的电子版**，可以很方便地进行修改和重新发布。如果您获得了 DezemingFamily 的系列电子书，可以从我们的网站 [https://dezeming.top/] 找到最新的版本。对文章的内容建议和出现的错误也欢迎在网站留言。

目录

一　基础的 LK 算法

稀疏光流法仅仅只跟踪图像中的部分点，而非全部点，因为可以通过捕获的特征点来跟踪，所以快速且可靠。而且因其将注意力只放在容易跟踪的特征点上，所以计算成本远远低于稠密跟踪。

论文 [1] 是 LK 光流法的开山之作，虽然一开始是试图计算稠密光流，但该方法很容易扩展到图像中的部分点集而非整幅图像，所以成为了稀疏光流法中的重要技术。它之所以可以用在稀疏光流中，主要是因为它仅仅依赖于点的局部信息来追踪，但如果像素点运动幅度过大，离开了局部窗口，那么就会跟踪失败。为了使得能够跟踪到较大幅度的运动，诞生了金字塔-LK 算法，这也是目前最常用的 LK 光流法的扩展，在图像金字塔上，从最低细节到最高细节都追踪，效果更加鲁棒。

在阅读本文之前，应该先了解《Horn-Schunck(HS) 光流法》中的基本内容。LK 光流法基于三个假设：

- 运动物体的色彩/亮度在很短的间隔时间内保持不变。
- 给定邻域内的速度向量场变化是缓慢的。
- 场景中相同表面的相邻点具有相似的运动，并且其投影到图像平面上的距离也比较近。

第三条假设与 HS 光流估计有点不同，因此得到了不同的关系式。

基于前两个假设，我们可以得到（原理同 HS 光流法）：

$$E_x u + E_y v + E_t = 0 \tag{一.1}$$

写为矩阵形式：

$$\begin{bmatrix} E_x & E_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -E_t \tag{一.2}$$

简写为：

$$\nabla E^T \cdot \mathbf{d} = -E_t \tag{一.3}$$

加上第三条假设：假设在 $m \times m = n$ 大小的窗内，光流是一个恒定值，就可以得到：

$$\begin{bmatrix} E_{x_1} & E_{y_1} \\ E_{x_2} & E_{y_2} \\ \cdots & \\ E_{x_n} & E_{y_n} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -E_{t_1} \\ -E_{t_2} \\ \cdots \\ -E_{t_n} \end{bmatrix} \tag{一.4}$$

在这个小的局部区域内采样最小二乘法，就可以求解出合适的 u 和 v 了。在时间间隔比较小的时候，光流速度就是位移。

但是如果相邻帧之间，第二帧的关键点离开了这个局部区域，那么该算法就不好使了，需要一些误差判断。比较理想的方式是使用图像金字塔进行追踪。

二　金字塔 LK 光流法简介

LK 算法的三个约束要求都不是很容易满足，但是，如果把图像放缩一下，比如原图中速度为 $\mathbf{d} = [u\ v] = [20\ 10]$，假设窗大小是 15×15，就会漏掉目标。图像缩小一倍以后，在该图的速度就变为了 $\mathbf{d} = [u\ v] = [10\ 5]$，则就被包含在窗里了。在图像金字塔中，上层金字塔分辨率低，下层金字塔分辨率高，设原始图像是第 0 层。假设在原始图像上某点位移为 $\mathbf{d}$，那么在每层的位移就可以写为：

$$\mathbf{d}_L = \frac{\mathbf{d}}{2^L} \tag{二.1}$$

我们假设图像 P 和 Q 是相邻两幅同样大小的图像（本来想用下标 t 和 $t+1$ 来表示的，但是下标 t 比如 E_t 已经被用来表示图像在时序的偏导了），我们需要衡量图像 P 的点 $P(\mathbf{p})$ 和图像 Q 的点 $Q(\mathbf{q})$ 之

间的相似关系，由此设定最小化目标（每一层的最小化目标都是如此，准确的光流值等于估计值加残差）：

$$\begin{aligned}\epsilon(\mathbf{d}) &= \epsilon(d_x, d_y, d_{xx}, d_{xy}, d_{yx}, d_{yy}) \\ &= \sum_{x=-\omega_x}^{\omega_x} \sum_{y=-\omega_y}^{\omega_y} \Big[P\big([x\ y]^T + \mathbf{p}\big) - Q\big([x\ y]^T + \mathbf{d} + \mathbf{p}\big)\Big]^2\end{aligned} \tag{二.2}$$

但是，有可能前一帧的点 $\mathbf{p}$ 所在的局部区域到 Q 上后不再是一个矩形窗，于是可以假设仿射变换矩阵 $\mathbf{A}$，用来描述 P 上的某一点 $\mathbf{p}$ 所在的局部区域映射到 Q 上的对应关系：

$$\begin{aligned}\epsilon(\mathbf{d}, \mathbf{A}) &= \epsilon(d_x, d_y, d_{xx}, d_{xy}, d_{yx}, d_{yy}) \\ &= \sum_{x=-\omega_x}^{\omega_x} \sum_{y=-\omega_y}^{\omega_y} \Big[P\big([x\ y]^T + \mathbf{p}\big) - Q\big(A[x\ y]^T + \mathbf{d} + \mathbf{p}\big)\Big]^2\end{aligned} \tag{二.3}$$

其中：

$$\mathbf{A} = \begin{bmatrix} 1 + d_{xx} & d_{xy} \\ d_{yx} & 1 + d_{yy} \end{bmatrix} \tag{二.4}$$

上面的两个求和函数构成了一个窗，不过在 P 上该窗是矩形的，但映射到 Q 上以后该窗就不一定是矩形区域了。但是当仿射变换矩阵 $\mathbf{A}$ 中 d_{xy} 和 d_{yx} 都是 0，那么矩形窗口变换到另一幅图以后还是矩形窗口。我们后面的描述中忽略矩阵 $\mathbf{A}$。

把最小化目标公式求导，再通过化简，最终得到最优的光流距离公式：

$$\begin{aligned}&\mathbf{d}_{opt} = G^{-1} b \\ &G = \sum_{x=-\omega_x}^{\omega_x} \sum_{y=-\omega_y}^{\omega_y} \begin{bmatrix} P_x^2([x\ y]^T + \mathbf{p}) & P_x I_y([x\ y]^T + \mathbf{p}) \\ P_x I_y([x\ y]^T + \mathbf{p}) & P_y^2([x\ y]^T + \mathbf{p}) \end{bmatrix} \\ &b = \sum_{x=-\omega_x}^{\omega_x} \sum_{y=-\omega_y}^{\omega_y} \begin{bmatrix} P_t([x\ y]^T + \mathbf{p}) P_x([x\ y]^T + \mathbf{p}) \\ P_t([x\ y]^T + \mathbf{p}) P_x I_y([x\ y]^T + \mathbf{p}) \end{bmatrix}\end{aligned} \tag{二.5}$$

这样就能计算出每一层的光流估计，然后移动到下一层上。求导的过程使用了泰勒展开，泰勒展开的前提是对应点位移变化比较小，而如何保证位移变化要尽可能小，则牵扯到“平移”操作，简单来说，就是上层在局部窗内检测到一个关键点位移以后，作用到下层时，就将下层的对应的图像进行平移，使得关键点位置更接近，这样就可以使用泰勒展开了。

由于顶层图像尺寸很小，设顶层的光流估计值 $\mathbf{g} = [0\ 0]$（叫估计值不太妥当，其实就是设置一个初始值），实际光流值 $\mathbf{d} = \mathbf{g} + \mathbf{d}'$。假设一共 m 层金字塔，我们用 0 到 m 下标来表示第几层的某个关键点的位移。移动时需要将移动向量扩大两倍，对于第 m 层，使用二.5求出来的结果是 $\mathbf{d}'_m$。

移动到第 $m-1$ 层以后，此时的光流估计值应该是：

$$\mathbf{g}_{m-1} = 2(\mathbf{g}_m + \mathbf{d}'_m) = 2\mathbf{d}'_m \tag{二.6}$$

我们先把 $m-1$ 层的图 P 先平移 $\mathbf{g}_{m-1}$ 个像素，它与 Q 的对应重叠区域（虽然理解容易但是不是很好描述）求出此时的新的光流结果 $\mathbf{d}'_{m-1}$，此时，总的位移就是：

$$2\mathbf{d}'_m + \mathbf{d}'_{m-1} \tag{二.7}$$

然后再向下移动，此时光流估计值是：

$$\mathbf{g}_{m-2} = 2(\mathbf{g}_{m-1} + \mathbf{d}'_{m-1}) = 4\mathbf{d}'_m + 2\mathbf{d}'_{m-1} \tag{二.8}$$

把 $m-2$ 层的图 P 先平移 $\mathbf{g}_{m-2}$ 个像素，它与 Q 的对应重叠区域求出此时的新的光流结果 $\mathbf{d}'_{m-2}$，该过程迭代重复，直到迭代到最后一层图像上。最终得到的光流值就是所有层的分段光流值的叠加：

$$\mathbf{d} = \sum_{L=0}^{m} 2^L \mathbf{d}'_L \tag{二.9}$$

当然也会有一些关键点是从中间层中开始出现的，这样出现的层设置初始值为 $\mathbf{g} = [0\ 0]$，仍然是一直往下迭代即可。

即使是能够匹配到点，也有可能匹配是错误的，所以需要能够排查掉错误点（比如 RANSAC 方法）。论文 [6] 和 [7] 中提出了用于跟踪光流的好的特征点获取方法，包括中值流跟踪，大家可以参考一下。我们后续也会在其他文章中介绍。

参考文献

[1] Lucas, B. D., & Kanade, T. (1981, August). An iterative image registration technique with an application to stereo vision. In IJCAI'81: 7th international joint conference on Artificial intelligence (Vol. 2, pp. 674-679).

[2] https://zhuanlan.zhihu.com/p/105998058

[3] Bouguet, Jean-Yves. "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm." Intel corporation 5.1-10 (2001): 4.

[4] https://www.bilibili.com/read/cv4874665?from=search

[5] https://blog.csdn.net/sgfmby1994/article/details/68489944

[6] Shi, Jianbo. "Good features to track." 1994 Proceedings of IEEE conference on computer vision and pattern recognition. IEEE, 1994.

[7] Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas. "Forward-backward error: Automatic detection of tracking failures." 2010 20th international conference on pattern recognition. IEEE, 2010.