Lucas-Kanade 光流法

光流（optical flow）是空间运动物体在观察成像平面上的像素运动的瞬时速度。光流法是利用图像序列中像素在时间域上的变化以及相邻帧之间的相关性来找到上一帧跟当前帧之间存在的对应关系，从而计算出相邻帧之间物体的运动信息的一种方法。可用于目标跟踪等领域。

光流法基本原理

直接法是从光流演变而来的。它们非常相似，具有相同的假设条件。光流描述了像素在图像中的运动，而直接法则附带着一个相机运动模型。为了说明直接法，我们先来介绍一下光流。

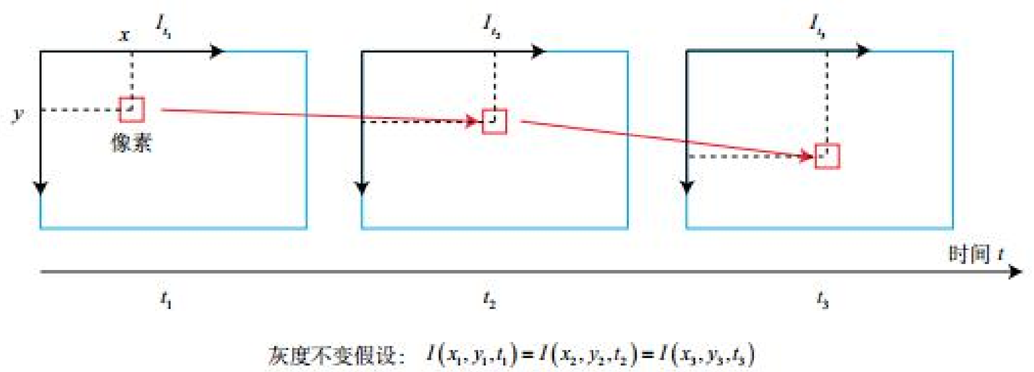


图 8-1 LK 光流法示意图。

光流是一种描述像素随着时间，在图像之间运动的方法，如图 8-1 所示。随着时间的经过，同一个像素会在图像中运动，而我们希望追踪它的运动过程。计算部分像素运动的称为**稀疏光流**，计算所有像素的称为**稠密光流**。稀疏光流以 Lucas-Kanade 光流为代表，并可以在 SLAM 中用于跟踪特征点位置。因此，本节主要介绍 Lucas-Kanade 光流，亦称 LK 光流。

在 LK 光流中，我们认为来自相机的图像是随时间变化的。图像可以看作时间的函数：$\boldsymbol{I}(t)$。那么，一个在 t 时刻，位于 (x, y) 处的像素，它的灰度可以写成

$$\boldsymbol{I}(x, y, t).$$

这种方式把图像看成了关于位置与时间的函数，它的值域就是图像中像素的灰度。现在考虑某个固定的空间点，它在 t 时刻的像素坐标为 x, y。由于相机的运动，它的图像坐标将发生变化。我们希望估计这个空间点在其他时刻里图像的位置。怎么估计呢？这里要引入光流法的基本假设：

灰度不变假设：同一个空间点的像素灰度值，在各个图像中是固定不变的。

对于 t 时刻位于 (x, y) 处的像素，我们设 $t + \mathrm{d}t$ 时刻，它运动到 $(x + \mathrm{d}x, y + \mathrm{d}y)$ 处。由于灰度不变，我们有：

$$\boldsymbol{I}(x + \mathrm{d}x, y + \mathrm{d}y, t + \mathrm{d}t) = \boldsymbol{I}(x, y, t). \tag{8.1}$$

灰度不变假设是一个很强的假设，实际当中很可能不成立。事实上，由于物体的材质不同，像素会出现高光和阴影部分；有时，相机会自动调整曝光参数，使得图像整体变亮或变暗。这些时候灰度不变假设都是不成立的，因此光流的结果也不一定可靠。然而，从另一方面来说，所有算法都是在一定假设下工作的。如果我们什么假设都不做，就没法设计实用的算法。所以，暂且让我们认为该假设成立，看看如何计算像素的运动。

对左边进行泰勒展开，保留一阶项，得：

$$\boldsymbol{I}\left(x + \mathrm{d}x, y + \mathrm{d}y, t + \mathrm{d}t\right) \approx \boldsymbol{I}\left(x, y, t\right) + \frac{\partial \boldsymbol{I}}{\partial x}\mathrm{d}x + \frac{\partial \boldsymbol{I}}{\partial y}\mathrm{d}y + \frac{\partial \boldsymbol{I}}{\partial t}\mathrm{d}t. \tag{8.2}$$

因为我们假设了灰度不变，于是下一个时刻的灰度等于之前的灰度，从而

$$\frac{\partial \boldsymbol{I}}{\partial x}\mathrm{d}x + \frac{\partial \boldsymbol{I}}{\partial y}\mathrm{d}y + \frac{\partial \boldsymbol{I}}{\partial t}\mathrm{d}t = 0. \tag{8.3}$$

两边除以 $\mathrm{d}t$，得：

$$\frac{\partial \boldsymbol{I}}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial \boldsymbol{I}}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} = -\frac{\partial \boldsymbol{I}}{\partial t}. \tag{8.4}$$

其中 $\mathrm{d}x/\mathrm{d}t$ 为像素在 x 轴上运动速度，而 $\mathrm{d}y/\mathrm{d}t$ 为 y 轴速度，把它们记为 u, v。同时 $\partial \boldsymbol{I}/\partial x$ 为图像在该点处 x 方向的梯度，另一项则是在 y 方向的梯度，记为 $\boldsymbol{I}_x, \boldsymbol{I}_y$。把

图像灰度对时间的变化量记为 $\boldsymbol{I}_t$，写成矩阵形式，有：

$$\begin{bmatrix} \boldsymbol{I}_x & \boldsymbol{I}_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\boldsymbol{I}_t. \tag{8.5}$$

我们想计算的是像素的运动 u, v，但是该式是带有两个变量的一次方程，仅凭它无法计算出 u, v。因此，必须引入额外的约束来计算 u, v。在 LK 光流中，我们假设**某一个窗口内的像素具有相同的运动**。

考虑一个大小为 $w \times w$ 大小的窗口，它含有 w^2 数量的像素。由于该窗口内像素具有同样的运动，因此我们共有 w^2 个方程：

$$\begin{bmatrix} \boldsymbol{I}_x & \boldsymbol{I}_y \end{bmatrix}_k \begin{bmatrix} u \\ v \end{bmatrix} = -\boldsymbol{I}_{tk}, \quad k = 1, \ldots, w^2. \tag{8.6}$$

记：

$$\boldsymbol{A} = \begin{bmatrix} [\boldsymbol{I}_x, \boldsymbol{I}_y]_1 \\ \vdots \\ [\boldsymbol{I}_x, \boldsymbol{I}_y]_k \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} \boldsymbol{I}_{t1} \\ \vdots \\ \boldsymbol{I}_{tk} \end{bmatrix}. \tag{8.7}$$

于是整个方程为：

这是一个关于 u, v 的超定线性方程，传统解法是求最小二乘解。最小二乘在很多时候都用到过：

$$\begin{bmatrix} u \\ v \end{bmatrix}^* = -\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{b}. \tag{8.9}$$

实验结果：

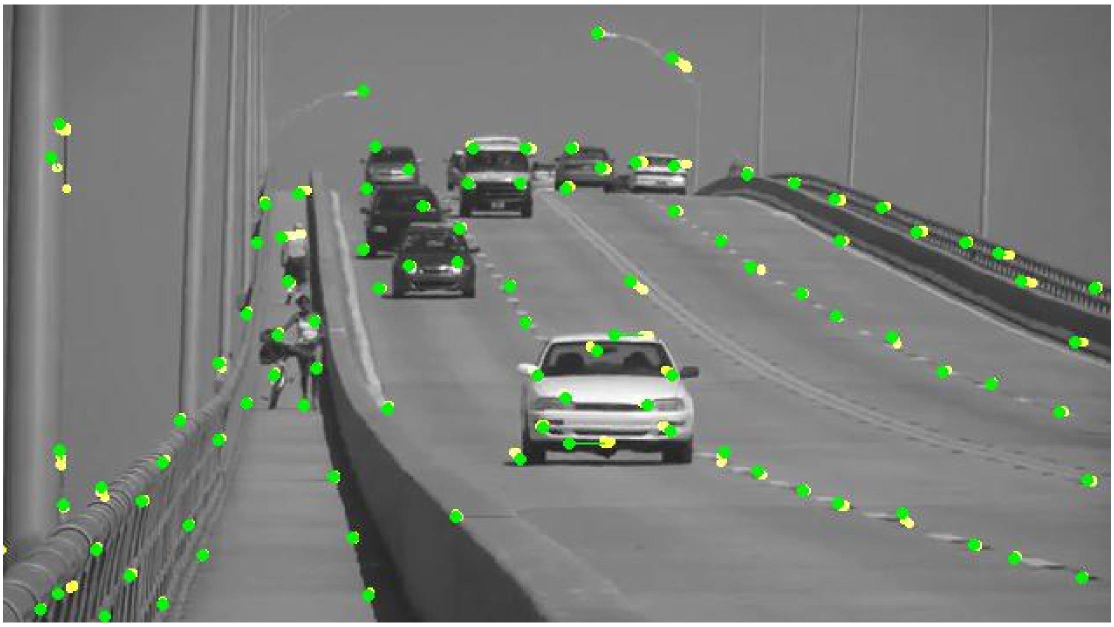
使用 LK 光流法对车流视频进行特征点跟踪。

1. 使用 GFTT 算法提取汽车的特征点；

2. 使用 LK 光流法跟踪下一帧特征点的位置；

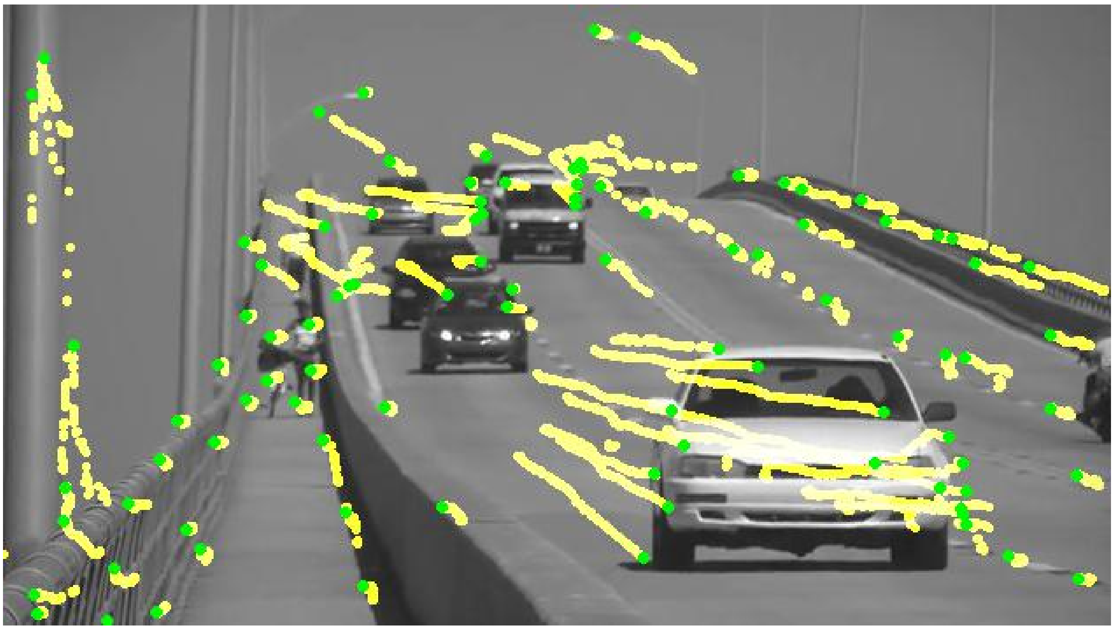
3. 画出特征点的轨迹；

以下截取的 4 幅图展示了汽车特征点的光流跟踪效果。绿色点表示特征点，黄色线条表示特征点的运动轨迹。

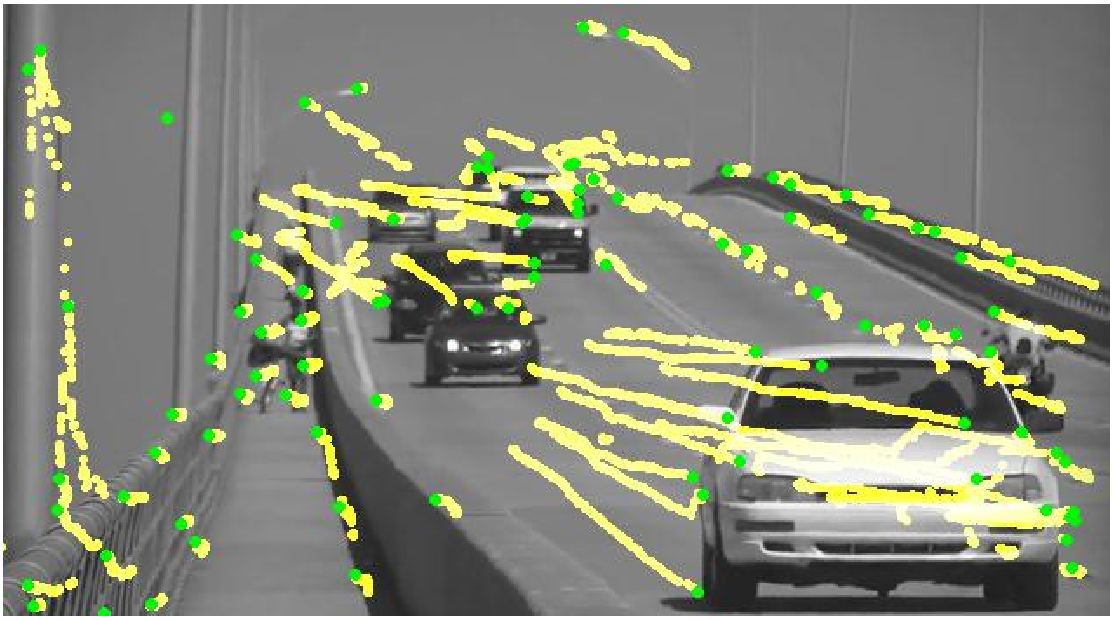


图（1）

图（2）



图（3）



图（4）