



本科生 《计算机视觉》 光流

王田

wangtian@buaa.edu.cn

202309



- 光流(Optical Flow)是空间运动物体在观察成像平面上**像素运动的瞬时速度**。
- 根据是否选取图像稀疏点进行光流估计，可以将光流估计分为**稀疏光流**和**稠密光流**。

左图为稀疏光流，选取一些特征明显的点进行光流估计和跟踪；右图为稠密光流，描述每个像素向下一帧运动大小和方向，映射到HSV颜色空间进行可视化。



- 根据定义，光流法实际上是解算观察平面上每一点的速度，对于二维平面，任一点速度可由 (u, v) 表示，称为**光流矢量**。其中， u 是水平方向的速度分量， v 是垂直方向的速度分量。
- 光流法假设同一目标在经过短暂时间 Δt 的前后亮度恒定，且位移不会发生剧烈变化，即：
 - 1.亮度恒定：场景中像素的亮度在相邻帧之间基本不变
 - 2.时间持续性：时间的变化不会引起像素位置的剧烈变化

根据光流的基本假设，对于 t 时刻位于 (x,y) 的物体像素，在 $(t+dt)$ 时刻位于 $(x+dx,y+dy)$ ，两个时刻该物体像素的亮度 I 相等：

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

对上式右端泰勒展开并略去高阶无穷小项：

$$I(x, y, t) = I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt$$
$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0$$

已知 $u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$ ，设 $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$, $I_t = \frac{\partial I}{\partial t}$

可以得到基本约束方程：

$$I_x u + I_y v + I_t = 0$$
$$\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -I_t$$

I_x, I_y, I_t 均可由图像数据求得，还需一个额外的约束条件可求解 (u, v)

■ Lucas-Kanade算法 (LK法) 是1981年由Bruce D. Lucas和Takeo Kanade提出的稀疏光流算法。

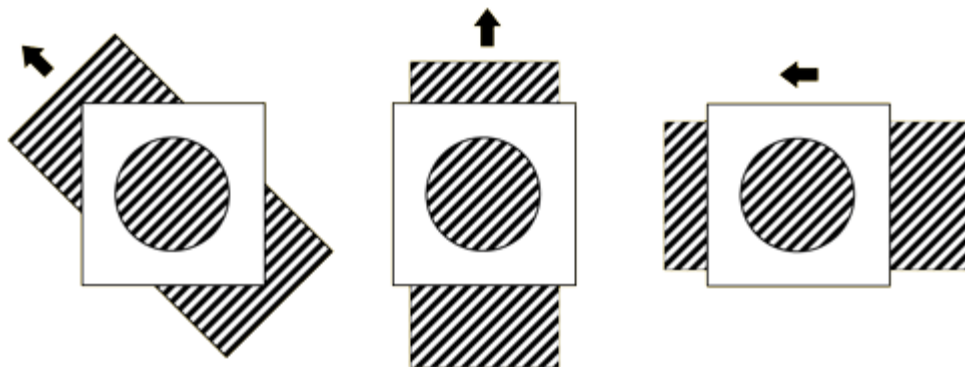
该方法在基本约束方程的基础上，提出**空间一致性**假设：像素点 $X = (x, y)$ 为中心的指定大小邻域内所有像素点光流值一致。对于 $n \times n$ 区域内所有点：

$$\begin{bmatrix} I_x & I_y \\ \vdots & \vdots \\ I_{x_n} & I_{y_n} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -I_{t_1} \\ \vdots \\ -I_{t_n} \end{bmatrix}$$

记 $A = \begin{bmatrix} I_x & I_y \\ \vdots & \vdots \\ I_{x_n} & I_{y_n} \end{bmatrix}$, $x = \begin{bmatrix} u \\ v \end{bmatrix}$, $b = \begin{bmatrix} -I_{t_1} \\ \vdots \\ -I_{t_n} \end{bmatrix}$, 若 $(A^T A)$ 可逆, 则可求解光流:

$$x = (A^T A)^{-1} A^T b$$

- 如果 $(A^T A)$ 不可逆，式将出现多解，即出现孔径问题（Aperture Problem）。如图，从圆孔中观察三种移动的条纹的变化，是一致的，从而无法通过圆孔得到条纹的真实移动方向（光流方向）。



- 因此采用角点检测算法选取像素点计算稀疏光流。

■ Farneback法是2003年由Gunner Farneback提出的稠密光流算法。

对灰度图像而言，像素点的灰度值可以看成二维变量的函数 $f(x, y)$ 。假设以感兴趣的像素点为中心，构建一个局部坐标系。对函数进行二项展开，可以近似为：

$$\begin{aligned} f(x, y) &\approx r_1 + r_2x + r_3y + r_4x^2 + r_5y^2 + r_6xy \\ &= \begin{pmatrix} x \\ y \end{pmatrix}^T \begin{pmatrix} r_4 & \frac{r_6}{2} \\ \frac{r_6}{2} & r_5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} r_2 \\ r_3 \end{pmatrix}^T \begin{pmatrix} x \\ y \end{pmatrix} + r_1 = \mathbf{X}^T \mathbf{A}_1 \mathbf{X} + \mathbf{b}_1^T \mathbf{X} + c_1 \end{aligned}$$

其中系数为据加权最小二乘法拟合邻近区域的信号值进行的估计， A 是对称矩阵， b 是向量， c 是标量。

设像素运动为 d ， f_1, f_2 分别代表了相邻两帧中同一像素前后的灰度函数，有：

$$\begin{aligned} f_2(X) &= f_1(X - d) \\ &= (X - d)^T A_1 (X - d) + b_1^T (X - d) + c_1 \\ &= X^T A_1 X + (b_1 - 2A_1 d)^T X + d^T A_1 d - b_1^T d + c_1 \\ &= X^T A_2 X + b_2^T X + c_2 \end{aligned}$$

由于图像场景中像素的外观信息在帧间运动不变，可以得到对应系数相同，且光流变化满足非奇异要求的这一特性，可得光流：

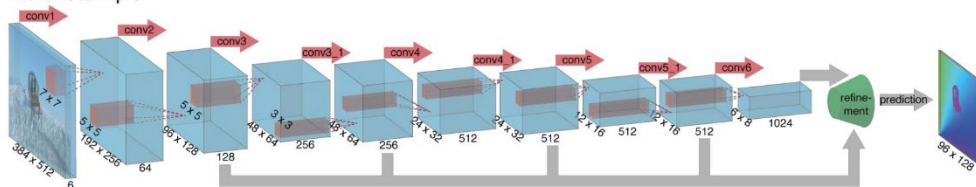
$$d = -\frac{1}{2} A_1^{-1} (b_2 - b_1)$$

■ FlowNet是第一个尝试使用卷积神经网络预测光流的工作。

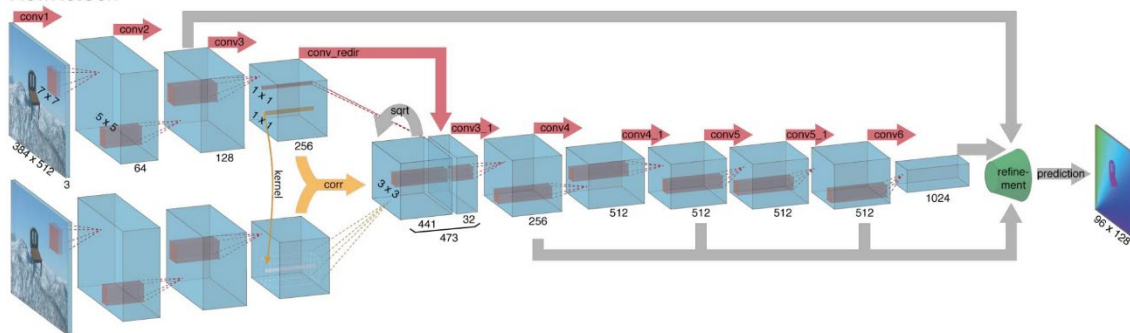
FlowNet分为FlowNetCorr、FlowNetSimple两部分：

FlowNetSimple简单地将两张图排列在一起直接输入网络，FlowNetCoor则先对两张图像分别进行卷积，获得较为高层的特征后再将信息合并进行运算

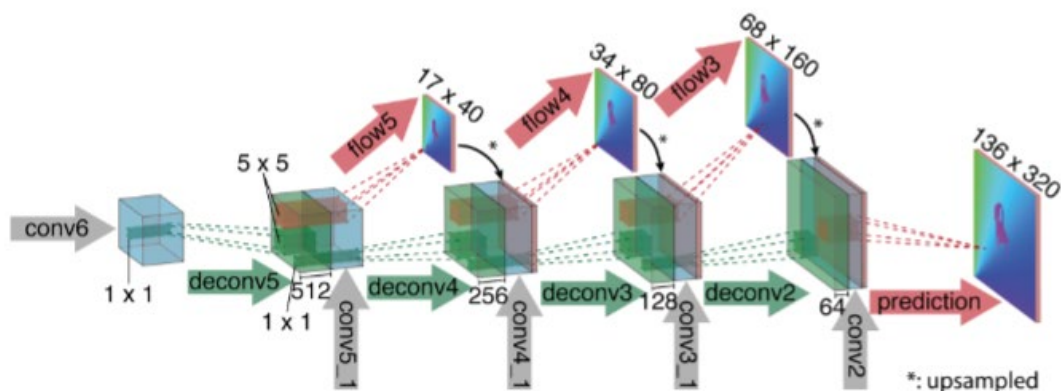
FlowNetSimple



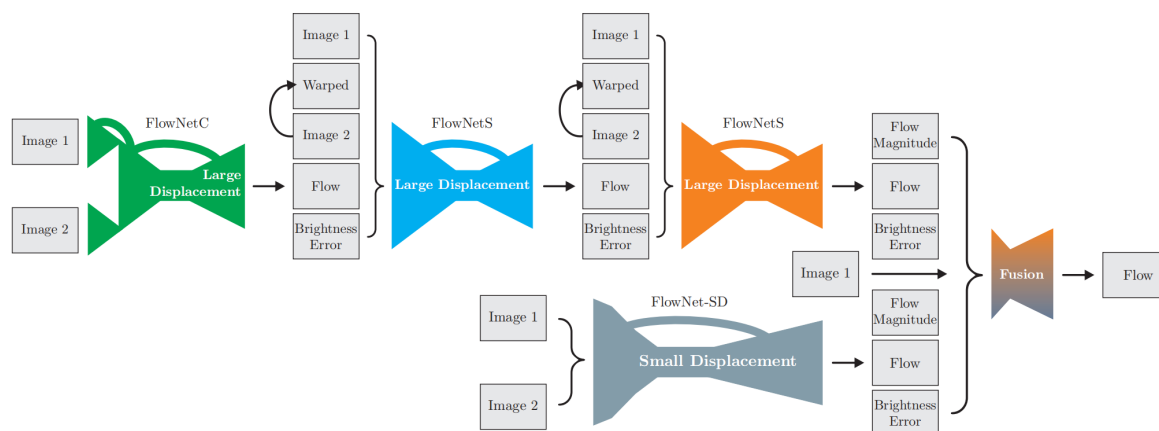
FlowNetCorr



- 在前期的网络中使用了池化网络，它可以聚合较大区域的信息，但是会导致分辨率降低。因此网络的最后通过扩张路径来实现refinement。
- 通过这种方式，既保留了来自粗糙特征图的高水平信息，又改善了低层特征图的局部信息。重复该过程四次，但最后的输出还是比原始大小小四倍。又由于更多的分辨率改善并不能提升效果，最后直接使用双线性上采样来恢复到原始输入大小。



- FlowNet2从性能上对 FlowNet 大幅优化，由若干子网络模块 (FlowNetC、FlowNetS、FlowNet-SD)组成，以应对不同物体的移动。
- 以FlowNetCorr为基础网络，额外堆叠两个FlowNetSimple模块。针对小位移情况改进了FlowNetSimple模块的结构，首先减小编码模块卷积核的尺寸和stride的大小以增加对小位移的分辨率。其次，在解码模块的反卷积层之间，均增加一层卷积层以便对小位移输出更加平滑的光流预测，命名为FlowNet-SD。



■ OpenCV

`cv2.calcOpticalFlowPyrLK()`

`cv2.calcOpticalFlowFarneback()`

■ MMFlow



- ✓ FlowNet (ICCV'2015)
- ✓ FlowNet2 (CVPR'2017)
- ✓ PWC-Net (CVPR'2018)
- ✓ LiteFlowNet (CVPR'2018)
- ✓ LiteFlowNet2 (TPAMI'2020)
- ✓ IRR (CVPR'2019)
- ✓ MaskFlowNet (CVPR'2020)
- ✓ RAFT (ECCV'2020)
- ✓ GMA (ICCV' 2021)