

# 报告2：光流估计

徐爽

数学与统计学院，统计学

邮箱: shuangxu@stu.xjtu.edu.cn

## 摘要

本文首先对LK光流, HS光流和LD光流模型进行了回顾。然后, 通过Matlab在Middlebury和天空之城数据集上对3中模型进行测试, 并对实验结果进行了简单的分析。本文涉及的代码、材料等见[https://github.com/xsxjtu/XJTU-CVPR\\_HOMEWORK\\_XuShuang](https://github.com/xsxjtu/XJTU-CVPR_HOMEWORK_XuShuang)。

## 1 光流

目标追踪等计算机视觉任务的一个目标是获得物体的运动场, 即物体在真实世界的运动在图像上的速度矢量投影. 然而, 物体的运动场通常无法直接获得. 1950年, Gibson提出了光流场的概念, 光流场是真实世界的物体运动在图像上的亮度投影. 在理想情况下, 光流场和运动场是一致的. 所以, 我们可以通过估计光流场来得到物体的运动. 在本节, 我们介绍LK光流, HS光流和LDOF三种光流方法.

### 1.1 LK光流

光流(optical flow)是在图片中明显的亮度模式的视觉运动。在理想状况下, 光流和运动场的相同的。在光流中, 一个基本假设是灰度不变假设, 即同一个空间点的像素灰度值, 在各个图像中是固定不变的。我们用 $I(x, y, t)$ 表示在时刻 $t$ 在于 $(x, y)$ 位置的像素。若它在 $t + dt$ 时刻运动到 $(x + dx, y + dy)$ 处, 这条假设可以由如下方程描述:

$$I(x + dx, y + dy, t + dt) = I(x, y, t). \quad (1)$$

对等式左边进行泰勒展开, 我们有,

$$I(x + dx, y + dy, t + dt) \approx I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt. \quad (2)$$

由于灰度是不变的, 所以

$$0 \approx \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt. \quad (3)$$

进而我们有,

$$-\frac{\partial I}{\partial t} \approx \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt}. \quad (4)$$

这里, 记像素在 $x, y$ 轴上的运动速度 $\frac{dx}{dt}, \frac{dy}{dt}$ 为 $u, v$ , 记图像在 $x, y, t$ 轴上的梯度为 $I_x, I_y, I_t$ , 则上式可以改写为

$$[I_x, I_y] \begin{bmatrix} u \\ v \end{bmatrix} = -I_t. \quad (5)$$

上面的等式关于 $u, v$ 是不适当的，即无法求得 $u, v$ 。所以需要再引入一定的约束。

一种方法是，加入空间一致性约束：假设在一个窗口中，所有像素具有相同的运动。若窗口大小为 $w \times w$ ，则我们可以获得 $w^2$ 个方程。记

$$\mathbf{A} = \begin{bmatrix} I_{x1} & I_{y1} \\ \vdots & \vdots \\ I_{xw^2} & I_{yw^2} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} I_{t1} \\ \vdots \\ I_{tw^2} \end{bmatrix} \quad (6)$$

当前，问题转化为

$$\mathbf{A}(u, v)^T = -\mathbf{b}. \quad (7)$$

该方程可以使用最小二乘法进行求解，即 $(u, v)^T = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A} \mathbf{b}$ 。

另一种方法是，加入局部空间一致性约束。即对像素 $(x, y)$ 的邻居 $\Omega$ 进行加权，最小化如下目标函数：

$$E(u, v) = \sum_{(x, y) \in \Omega} w(x, y)(I_x u + I_y v + I_t)^2 = (u, v, 1) \mathbf{A}(u, v, 1)^T, \text{ with } \mathbf{M} = \sum_{(x, y) \in \Omega} w(x, y) \begin{pmatrix} I_x^2 & I_x I_y & I_x I_t \\ I_x I_y & I_y^2 & I_y I_t \\ I_x I_t & I_y I_t & I_t^2 \end{pmatrix} \quad (8)$$

对目标函数关于 $(u, v)$ 求导，并令其为零，可得

$$\mathbf{G} \odot \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} (u, v)^T = -\mathbf{G} \odot (I_x I_t, I_y I_t)^T \quad (9)$$

其中 $\odot$ 为卷积操作， $\mathbf{G}$ 是由权重 $w(x, y)$ 决定的卷积核。使用线性代数可以得到 $(u, v)$ 的解。

## 1.2 HS光流

HS光流仍然假设灰度不变，所以公式

$$I_x u + I_y v + I_t = 0 \quad (10)$$

仍然成立。此外，HS光流还假设在局部区域内速度 $u, v$ 的变化不大。这个假设称为速度平滑假设。通常，记速度平滑项为

$$\xi^2 = \left( \frac{du}{dx} \right)^2 + \left( \frac{du}{dy} \right)^2 + \left( \frac{dv}{dx} \right)^2 + \left( \frac{dv}{dy} \right)^2 \quad (11)$$

HS的目标函数为

$$L = \iint (I_x u + I_y v + I_t)^2 + \alpha \xi^2 dx dy, \quad (12)$$

这里的 $\alpha$ 为超参数，平衡速度光滑约束和灰度不变约束的强弱。

## 1.3 LD光流

在2011年，Brox和Malik提出了Large Displacement Optical Flow (LDOF)。LDOF是传统光流计算最先进的技术之一。其目标函数包含了5项。首先，根据灰度不变假设，我们有灰度的损失函数

$$E_{color}(\mathbf{w}) = \int_{\Omega} \Psi(|M_2(\mathbf{x} + \mathbf{w}(\mathbf{x})) - M_1(\mathbf{x})|^2) d\mathbf{x}. \quad (13)$$

这里,  $M_i$ 表示第*i*张图像,  $\mathbf{x} = (x, y)^T$ 表示图像域 $\Omega$ 中的一点,  $\mathbf{w} = (u, v)^T$ 表示光流场, 即函数 $\mathbf{w} : \Omega \rightarrow \mathbb{R}^2$ .  $\Psi(s^2) = \sqrt{s^2 + \epsilon^2}$ 是 $L_1$ 范数, 为了增加数值稳定性, 令 $\epsilon = 0.001$ . 与LK和HS光流相比, LDOF的灰度损失函数可以增强鲁棒性. 由于光线的变化, 灰度假设不总是可靠的. 因此LDOF加入了梯度约束, 来保证模型对可加亮度的不变性, 即

$$E_{grad}(\mathbf{w}) = \int_{\Omega} \Psi(|\nabla M_2(\mathbf{x} + \mathbf{w}(\mathbf{x})) - \nabla M_1(\mathbf{x})|^2) d\mathbf{x}. \quad (14)$$

此外, 与HS光流类似, LDOF还加入了速度光滑性的约束,

$$E_{smooth}(\mathbf{w}) = \int_{\Omega} \Psi(|\nabla u(\mathbf{x}) + \nabla v(\mathbf{x})|^2) d\mathbf{x}. \quad (15)$$

除了上述的3个约束, LDOF还考虑了特征匹配的速度光滑假设: 若两张图像存在特征匹配, 那么这两点的速度相同. 匹配光滑假设可以由下面的公式描述

$$E_{match}(\mathbf{w}) = \int \delta(\mathbf{x}) \rho(\mathbf{x}) \Psi(|\mathbf{w}(\mathbf{x}) - \mathbf{w}_1(\mathbf{x})|^2) d\mathbf{x}, \quad (16)$$

其中,  $\delta(\mathbf{x})$ 是指示函数, 表示点 $\mathbf{x}$ 是否存在特征匹配;  $\rho(\mathbf{x})$ 表示匹配的可信度;  $\mathbf{w}_1(\mathbf{x})$ 表示匹配点的速度. 匹配任务可以由下面的公式描述

$$E_{desc}(\mathbf{w}_1) = \int \delta(\mathbf{x}) |\mathbf{f}_2(\mathbf{x} + \mathbf{w}_1(\mathbf{x})) - \mathbf{f}_1(\mathbf{x})|^2 d\mathbf{x}. \quad (17)$$

这里 $\mathbf{f}_i(\mathbf{x})$ 表示第*i*张图像的特征向量场.

最终, LDOF的目标函数为

$$E(\mathbf{w}) = E_{color}(\mathbf{w}) + \gamma E_{grad}(\mathbf{w}) + \alpha E_{smooth}(\mathbf{w}) + \beta E_{match}(\mathbf{w}) + E_{desc}(\mathbf{w}_1) \quad (18)$$

## 2 实验

在本节中, 我们主要使用LK, HS, LDOF<sup>1</sup>算法进行两帧图像的光流实验. 我们使用了光流领域常用的 Middlebury 数据集<sup>2</sup>. 该数据集中的图像分辨率较低(不高于640x480), 场景比较简单. 因此, 我们在电视剧《天空之城(SKY castle)》<sup>3</sup>中截取了部分高分辨率(1080x720)复杂场景的图像进行测试. 实验使用了Matlab2018b, 在一台电脑(i7-8750 CPU @2.20GHz, 8G RAM, SSD)上运行.

图1-8分别展示了Middlebury和天空之城的结果. 可以看出LK模型的效果最差, 无法反映出真实的光流场; HS的表现较好, 能够大致反映物体的运动轨迹; LDOF的表现最好, 在大部分场景下都可以反映出真实的物体运动情况. 如图2所示, 在Middlebury(Backyard)中, 有4个小孩在跳跃, 1个小球在下落. LDOF可以完美地捕捉到每个物体的运动, LK只能捕捉到一些边缘信息. 虽然HS也能捕捉到每个物体的运动, 但是HS算法受到了背景噪声的影响. 如图4所示, 在Middlebury(Dumptruck)中, 有4辆车正在移动. LDOF可以检测到3辆车正在移动, 而另一辆车无法检测到; HS可以检测到全部车辆的移动.

天空之城数据集的场景更加复杂. 在该数据上, LDOF依然表现最好, HS和LK次之. 在天空之城(ep01)中, 摄像机保持不动, 有7个物体在移动: 2人鞠躬, 2人鼓掌, 3人走路. LDOF检测到了所有物体的移动, 然而HS错误地检测到了背景的移动(比如, 门框和墙). 在天空之城(ep02)中, 正在下雨, 物体保持不动, 摄像机在缓慢移动. 在这

<sup>1</sup><http://www.cs.cmu.edu/~katef/LDOF.html>

<sup>2</sup><http://vision.middlebury.edu/flow/data/comp/zip/eval-color-twoframes.zip>

<sup>3</sup><http://tv.jtbc.joins.com/skycastle>



图 1: Middlebury(Army)数据集的结果. 从左到右分别为: wrapped image, LDOF, LK, HS.



图 2: Middlebury(Backyard)数据集的结果. 从左到右分别为: wrapped image, LDOF, LK, HS.

张图像中, 物体可以根据与摄像机的距离被分为远中近3类, 近处的物体移动速度较快, 反之远处的物体移动较慢. 此时, LDOF无法检测到远处物体的移动, HS受到下雨的影响检测效果也较差. 在ep03和ep04中, LDOF能够取得较好的效果, HS次之.

### 3 总结

光流的研究历史比较长. 最早的LK光流方法仅仅使用了灰度不变和空间一致性假设, 且使用L2范数作为损失函数. 其鲁棒性较差, 在实际场景中的效果不佳. HS光流除了灰度不变假设, 还加入了速度平滑约束, 提升了LK光流的表现. 最近提出的LD光流是传统光流计算中效果最好的方法之一. LD光流一方面使用了L1范数作为损失函数, 另一方面考虑了5项约束, 大大提升了LK和HS光流的效果, 在实际场景中表现很好.

在我们的实验中发现, LK光流很多时候完全失效, LK光流则能检测到部分物体运动, 但是容易受到噪声影响, LDOF则非常鲁棒. 然而在部分场景中, LDOF仍然无法得到理想的效果.



图 3: Middlebury(Basketball)数据集的结果. 从左到右分别为: wrapped image, LDOF, LK, HS.



图 4: Middlebury(Dumptruck)数据集的结果. 从左到右分别为: wrapped image, LDOF, LK, HS.

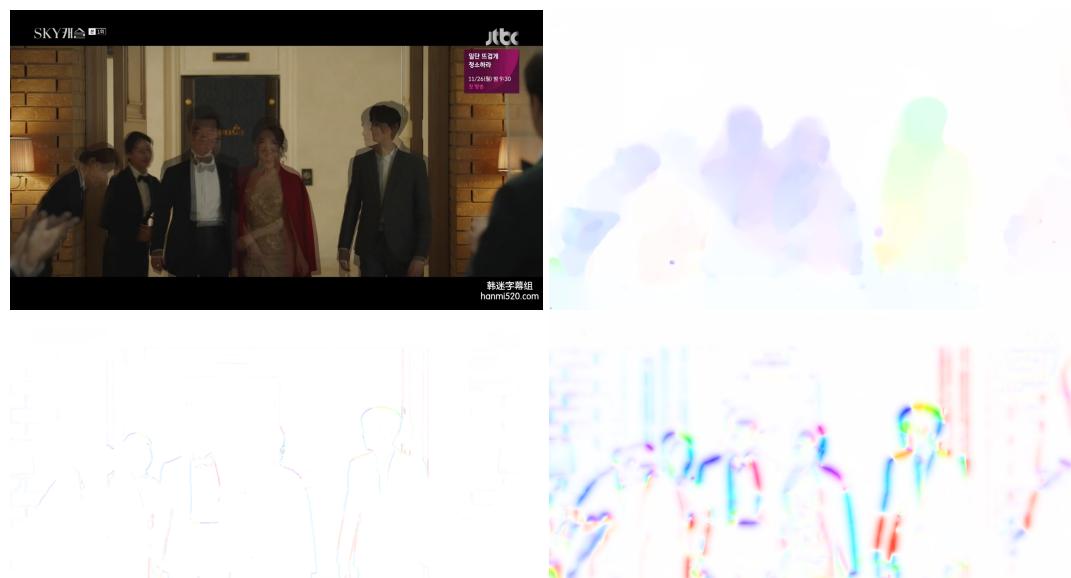


图 5: 天空之城(ep01)的结果. 左上: wrapped image, 右上: LDOF, 左下: LK, 右下: HS.



图 6: 天空之城(ep02)的结果. 左上: wrapped image, 右上: LDOF, 左下: LK, 右下: HS.

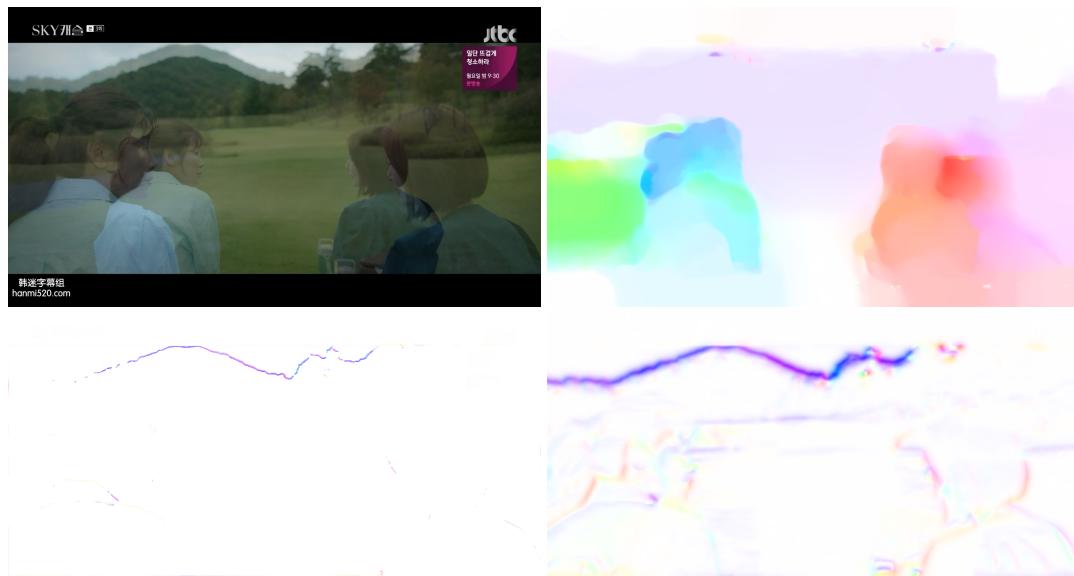


图 7: 天空之城(ep03)的结果. 左上: wrapped image, 右上: LDOF, 左下: LK, 右下: HS.



图 8: 天空之城(ep04)的结果. 左上: wrapped image, 右上: LDOF, 左下: LK, 右下: HS.