

## 运动目标分割实验报告

**摘要：**本实验的目的是从图像序列中检测运动目标，并将其分割。首先，本文介绍了运动目标检测常用的三种方法：背景差分法、帧间差分法、光流法。并对光流法的原理进行了详细介绍。本实验采用了帧间差分法和光流法（包括 Horn-Schunck 算法和 Lucas-Kanade 算法）实现运动目标分割。运用传统帧差法提取运动目标的过程包括：获取图片流进行平滑去噪，将其转换为灰度图像，然后通过帧差法得到二值化图像，最后再进行形态学去噪处理，进行标记得到结果。光流法的过程包括读入图片流，转化为灰度图像，分别用 Horn-Schunck 算法和 Lucas-Kanade 算法计算光流，光流场灰度化，对灰度化后的光流场进行二值化，形态学去噪得到更加准确的运动目标，标记图像得到结果。最后我们分析了实验结果，比较了帧间差分法和光流法实现运动目标分割的优缺点。

### 一、实验目的和要求

#### （1）实验目的

从图像序列中检测出“不同”的运动目标，将其分割出来，并用不同颜色加以标注。

#### （2）实验要求

利用给定数据中的一组或多组进行测试，给出一帧的分割结果。

### 二、实验方法和原理

运动目标检测方法发展到今天，已经有非常多的方法，然而许多方法都是在以下的三种常用的方法中改进而来。下面是运动目标检测技术常用的三种方法：

（1）光流法：光流是一种简单实用的图像运动的表达方式，光流是一个图像序列中的图像的亮度模式的表观运动，光流不但包括被观察物体的运动信息，而且还包括相关的结构信息。由于光流场具有不连续性，因此可以用来将研究的图像分割成对应不同的运动物体的区域。基于光流方法的运动目标检测具有运动目标随时间变化的光流特性，由上可知，光流中同时包含着被观察物体的运动信息和结构信息，然而在实际中由于遮挡，噪声，透明等原因，使得光流场的基本方程的假设的灰度守恒条件不能满足，从而不能得到正解，此外，大多数光流方法的计算复杂，只能得到稀疏的光流场，不适于实时处理。因此，对于实时性和精确性有要求的系统一般不会采用该方法。

（2）背景差分法：背景差分法在运动目标检测技术中是一种非常常用的方法，该方法是利用当前图像与背景图像进行差分，来检测出运动区域的。这种方法一般都能够提供较完整的特征数据。由于该方法不受运动目标速度的限制，因此能够较完整地提取出来运动目标，但是其检测性能与背景图像提取的好坏有很大的关联，并且对光照条件和外部条件造成的场景变化过于敏感，如果在非受控环境下，该算法还需要加入对背景图像的更新机制，并且该方法不适用于背景灰度有很大变化的情况。

（3）帧间差分法：帧间差分法是一种通过对视频图像序列中相邻两帧作差分运算来获得运动目标轮廓的方法。该方法可适用于存在多个运动目标和摄像机移动的情况。当所监控的场景中有异常的物体运动时，帧与帧之间就会出现较为明显的差别。通过两帧相减，得到两帧图像亮度差的绝对值，判断它是否大于阈

值,根据这个判断条件,我们可以得出图像的二值化图像,并以此来分析视频或图像序列的运动特性,确定图像序列中是否有物体运动。图像序列逐帧的差分,就相当于对图像序列进行了时域下的高通滤波。虽然帧间差分算法对环境有很好的适应性,并且差分图像受光线变化影响小,检测的结果有效而稳定,但是它只能检测相对运动的目标,检测出的目标位置不一定精确,并且不能提取出较完整的运动目标。该方法在较大程度上依赖差分帧的选择时机和目标的运动速度,因此也有一定的局限性。

### 光流法的原理介绍:

光流是指空间运动物体在观测成像面上的像素运动的瞬时速度,它利用图像序列像素强度数据的时域变化和相关性来确定各自像素位置的“运动”,即反映图像灰度在时间上的变化与景物中物体结构及其运动的关系。将二维图像平面特定坐标点上的灰度瞬时变化率定义为光流矢量。视觉心理学认为人与被观察物体发生相对运动时,被观察物体表面带光学特征的部位移动给人们提供了运动和结构的信息。当相机与场景目标间有相对运动时所观察到的亮度模式运动称之为光流(optical flow),或者说物体带光学特征部位的移动投影到视网膜平面(也即图像平面)上就形成了光流。

光流场是指图像灰度模式的表观运动,它是一个二维矢量场,所包含的信息就是各个像素点的瞬时运动速度矢量信息。

光流场每个像素都有一个运动矢量,因此可以反映相邻帧之间的运动<sup>[1]</sup>。

一般情况下,光流由相机运动,场景中目标运动或两者的共同运动产生。光流场的计算大致可分为三类:基于梯度的方法(Horn-Schunck);基于匹配的方法;基于能量的方法。

基于梯度的方法利用图像灰度的梯度来计算光流,是研究最多的方法。基于梯度的方法根据运动前后图像灰度保持不变这个基本假设,导出光流约束方程。由于光流约束方程并不能唯一的确定光流,因此需要导入其他的约束。根据引入的约束不同,基于梯度的方法又可以分为全局约束方法和局部约束方法。全局约束的方法假定光流在整个图像范围内满足一定的约束条件;而局部约束的方法假定在给定点周围的一个小区域内,光流满足一定的约束条件。

基于匹配的方法,这类方法是将速度  $v_m$  定义为视差  $d=(dx,dy)^T$ ,使得两个时刻的图像区域的匹配最佳。为了找到最佳匹配,我们可以对定义在  $d$  上的相似度量,如归一化的互相关系数,进行最大化,也可以对某一距离度量,如光强度差的平方和,进行最小化。

梯度光流法又分为全局约束方法和局部约束方法。全局约束方法假定光流在整个图像范围内满足一定的约束条件,而局部约束的方法假定在给定点周围的一个小区域内,光流满足于一定的约束条件。下面先导出光流约束方程。然后给出两种比较典型的基于梯度的方法。

假定图像上点  $m=(x,y)^T$  在时刻  $t$  的灰度值为  $I=(x,y,t)\sqrt{2}$ ,经过时间间隔  $dt$  后,对应点的灰度为  $I(x+dx,y+dy,t+dt)$ <sup>[3]</sup>,当时,可以认为两点的灰度不变,也就是:

$$I(x+dx,y+dy,t+dt)=I(x,y,t) \quad (2-1)$$

如果图像灰度随  $x,y,t$  缓慢变化,可以将(1)式左边泰勒级数展开:

$$I(x+dx,y+dy,t+dt)=I(x,y,t)+\varepsilon \quad (2-2)$$

其中  $\varepsilon$  代表二阶无穷小项。由于  $dt \rightarrow 0$ ，忽略  $\varepsilon$ ，可以得到：

$$\frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt = 0 \quad (2-3)$$

令  $u=dx$ ， $v=dy$  代表  $x,y$  方向上的光流， $I_x=\frac{\partial I}{\partial x}$ ， $I_y=\frac{\partial I}{\partial y}$ ， $I_t=\frac{\partial I}{\partial t}$  分别代表图

像灰度相对于  $x,y,t$  的偏导，式(2-3)可以写成：

$$I_x u + I_y v + I_t = 0 \quad (2-4)$$

此式即光流场的基本方程。写成向量形式，即：

$$V_m + I_t = 0 \quad (2-5)$$

其中  $I_t=(I_x, I_y)$  是图像在点  $m$  处的梯度， $V_m(U,V)$  是点  $m$  的光流。上式称为光流约束方程，是所有基于梯度的光流计算方法的基础。

### Horn-Schunck 经典光流场计算方法：

Horn-Schunck 算法提出了光流的平滑性约束。即：图像上任一点的光流并不是独立的，光流在整个图像范围内平滑变化。因此 Horn-Schunck 算法是一种全局约束的方法。

设平滑性约束项为极小化：

$$E_s = \iint (u_x^2 + u_y^2 + v_x^2 + v_y^2) dx dy \quad (2-6)$$

由基本等式，显然要求极小化：

$$E_c = \iint (I_x u + I_y v + I_t)^2 dx dy \quad (2-7)$$

于是，由(2-6)和(2-7)式可知，最后求得光流应满足 (2-8) 式：

$$\min \iint [\lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2) + (I_x u + I_y v + I_t)^2] dx dy \quad (2-8)$$

这里  $\lambda$  的取值要考虑图中的噪声情况，如果噪声较强，说明图像数据本身的置信度较低，需要更多的依赖光流约束，所以  $\lambda$  可以取较大的值；反之，取较小的值。

为了满足(2-8)，可将该式对  $u$  和  $v$  分别求导，并取导数为 0。这样就得到：

$$I_y^2 v + I_x I_y u = -\lambda^2 \nabla v - I_y I_t \quad (2-9)$$

$$I_x^2 v + I_x I_y u = -\lambda^2 \nabla v - I_y I_t \quad (2-10)$$

以上两式也称为 Euler 方程。如果令  $\bar{u}$  和  $\bar{v}$  分别表示  $u$  邻域和  $v$  邻域中的均值(可用图像平滑算子求得)，并令  $\nabla u = u - \bar{u}$  和  $\nabla v = v - \bar{v}$ ，则式(2-9)和(2-10)改写成：

$$I_t = \frac{1}{4}(I_{i,j,k+1} + I_{i+1,j,k+1} + I_{i,j+1,k+1} + I_{i+1,j+1,k+1}) - \frac{1}{4}(I_{i,j,k} + I_{i+1,j,k} + I_{i,j+1,k} + I_{i+1,j+1,k}) \quad (2-11)$$

$$(I_x^2 + \lambda^2) u + I_x I_y v = \lambda^2 \bar{u} - I_x I_t \quad (2-12)$$

$$(I_y^2 + \lambda^2) v + I_x I_y u = \lambda^2 \bar{v} - I_y I_t \quad (2-13)$$

从上式解得：

$$u = \bar{u} - \frac{I_x(I_x \bar{u} + I_y \bar{v} + I_t)}{\lambda^2 + I_x^2 + I_y^2} \quad (2-14)$$

$$v = \bar{v} - \frac{I_y(I_x \bar{u} + I_y \bar{v} + I_t)}{\lambda^2 + I_x^2 + I_y^2} \quad (2-15)$$

式(2-14)和(2-15)提供了用迭代法求解  $u$  和  $v$  的基础<sup>[7]</sup>。实际中，常用松弛迭方程进行求解：

$$u^{(k+1)} = \bar{u}^{(k)} - I_x \frac{I_x \bar{u}^{(k)} + I_y \bar{v}^{(k)} + I_t}{\lambda^2 + I_x^2 + I_y^2} \quad (2-16)$$

$$v^{(k+1)} = \bar{v}^{(k)} - I_y \frac{I_x \bar{u}^{(k)} + I_y \bar{v}^{(k)} + I_t}{\lambda^2 + I_x^2 + I_y^2} \quad (2-17)$$

其中  $k$  是循环数， $u^{(0)}$  和  $v^{(0)}$  是初始值，可以取为 0。 $\bar{u}$  和  $\bar{v}$  是局部平均， $\lambda$  为权重系数，根据导数求取的精确度确定。

在实际求解过程中，需要估计亮度的时间和空间微分。这可在图像点的一个  $2 \times 2 \times 2$  立方邻域中估计，如果下标  $i, j, k$  分别对应  $x, y, t$ ，那么 3 个一阶偏导分别是：

$$I_x = \frac{1}{4}(I_{i+1,j,k} + I_{i+1,j+1,k} + I_{i+1,j,k+1} + I_{i+1,j+1,k+1}) - \frac{1}{4}(I_{i,j,k} + I_{i,j+1,k} + I_{i,j,k+1} + I_{i,j+1,k+1}) \quad (2-18)$$

$$I_y = \frac{1}{4}(I_{i,j+1,k} + I_{i+1,j+1,k} + I_{i,j+1,k+1} + I_{i+1,j+1,k+1}) - \frac{1}{4}(I_{i,j,k} + I_{i+1,j,k} + I_{i,j,k+1} + I_{i+1,j,k+1}) \quad (2-19)$$

$$I_t = \frac{1}{4}(I_{i,j,k+1} + I_{i+1,j,k+1} + I_{i,j+1,k+1} + I_{i+1,j+1,k+1}) - \frac{1}{4}(I_{i,j,k} + I_{i+1,j,k} + I_{i,j+1,k} + I_{i+1,j+1,k}) \quad (2-20)$$

也就是用一阶差分来替代灰度  $I$  关于  $x, y, t$  轴的偏导。

上述算法的实现相对简单，计算复杂性较低。但是这种技术存在着严重缺陷。首先，图像灰度保持假设对于许多自然图像序列来讲都是不合适的，尤其是在图像的遮合边缘处和(或)当运动速度较高时，基于灰度保持假设的约束存在较大误差。其次，在图像的遮合区域，速度场是突变的，而总体平滑约束则迫使所估计的光流场平滑地穿过这一区域，此过程平滑掉了有关物体形状的非常重要的信息。第二，微分技术的一个要求是  $I(x, y, t)$  必须是可微的，这暗示着需对图像数据进行时空预平滑，以避免混叠效应；而且数值微分的求取具有病态性，如果处理不当将对最终的速度估计产生显著影响。

#### Lucas-Kanade 算法：

与 Horn 方法不同，Lucas-Kanade 方法是基于局部约束的。假定以  $p$  点为中

心的一个小区内各点的光流相同，对区域内不同的点给予不同的权重，这样光流的计算就转化为最小化如下的方程：

$$\sum_{x \in \Omega} W^2(x) [\nabla \mathbf{I}(x, t) \cdot \mathbf{v} + \mathbf{I}_t(x, t)]^2 \quad (2-21)$$

上式中， $\Omega$  代表以  $p$  点为中心的一个小的区域， $W(x)$  为窗函数，代表区域中各点的权重，离  $p$  点越近，权重越高。(2-21)的解可以由下面的方程得到：

$$\mathbf{A}^T \mathbf{W}^2 \mathbf{A} \mathbf{v} = \mathbf{A}^T \mathbf{W}^2 \mathbf{b} \quad (2-22)$$

对于邻域  $\Omega$  内的  $n$  个点  $m_i$ ，其中，

$$\mathbf{A} = (\nabla \mathbf{I}(\mathbf{x}_1), \nabla \mathbf{I}(\mathbf{x}_2), \dots, \nabla \mathbf{I}(\mathbf{x}_n))^T \quad (2-23)$$

$$\mathbf{W} = \text{diag}(\mathbf{W}(\mathbf{x}_1), \mathbf{W}(\mathbf{x}_2), \dots, \mathbf{W}(\mathbf{x}_n)) \quad (2-24)$$

$$\mathbf{b} = -(\mathbf{I}_t(\mathbf{x}_1), \mathbf{I}_t(\mathbf{x}_2), \dots, \mathbf{I}_t(\mathbf{x}_n))^T \quad (2-25)$$

最后，方程的解为：

$$\mathbf{v} = (\mathbf{A}^T \mathbf{W}^2 \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W}^2 \mathbf{b} \quad (2-26)$$

实际上， $\mathbf{A}^T \mathbf{W}^2 \mathbf{A}$  为  $2 \times 2$  矩阵：

$$\mathbf{A}^T \mathbf{W}^2 \mathbf{A} = \begin{bmatrix} \sum W^2(x) I_x^2(x) & \sum W^2(x) I_x(x) I_y(x) \\ \sum W^2(x) I_y(x) I_x(x) & \sum W^2(x) I_y^2(x) \end{bmatrix} \quad (2-27)$$

上式中所有的求和都是在  $\Omega$  的所有点上进行的。

假设  $\mathbf{A}^T \mathbf{W}^2 \mathbf{A}$  的特征值为  $\lambda_1$  和  $\lambda_2$ ，并且  $\lambda_1 \geq \lambda_2$ ，则：

- (1) 如果  $\lambda_1 > \tau, \lambda_2 > \tau$ ，则利用(2-27)计算  $\mathbf{v}$ ；
- (2) 如果  $\lambda_1 > \tau, \lambda_2 < \tau$ ，则不能得到光流的完整信息；
- (3) 如果  $\lambda_1 < \tau$ ，则认为数据不可靠，不能计算光流。

在 Lucas-Kanade 算法的具体实现中， $\Omega$  为  $3 \times 3$ ，则可以得到一个超定的图像流约束方程：

$$\begin{bmatrix} I_{x1} & I_{x2} & \dots & I_{x9} \\ I_{y1} & I_{y2} & \dots & I_{y9} \end{bmatrix}^T \begin{bmatrix} u & v \end{bmatrix}^T = \begin{bmatrix} -I_{t1} & -I_{t2} & \dots & -I_{t9} \end{bmatrix}^T \quad (2-28)$$

图像流约束方程实际是速度平面  $(u, v)$  上的直线方程，如果考虑图像序列中连续的  $J(J \geq 2)$  帧图像，并假定目标的运动速度在  $J$  帧图像里近似保持不变，对于运动目标而言，其在连续  $J$  帧图像里的  $J$  条运动约束直线，必在速度平面近似交于一点。

为了进一步提高 Lucas-Kanade 方法的准确度以及运算速度，在实际应用中，又将高斯金字塔分层与 Lucas-Kanade 方法结合起来，采用了由粗到精的分层策略将图像分解成不同的分辨率，随着级别的增加，分辨率越来越低，并将在粗尺度下得到的结果作为下一尺度的初始值，在不同的分辨率上对图像序列进行流速计算，这是计算大的运动速度的有效的技术手段。

基于微分法的光流计算方法仅适用于小图像运动的问题，金字塔光流法可以

解决此问题。该方法的基本思想是构造图像序列的一个金字塔，较高的层是下层平滑后的下采样形式，原始图像层数等于零。当图像分解到一定的层后，相邻帧间图像运动量将变得足够小，满足光流计算的约束条件，可以直接进行光流估计。在实际计算时，由高层到低层进行，当某一级的光流增量计算出来后，将加到其初始值上，再进行投影重建，作为其下一层的光流计算初值。这一过程不断进行，直至估计出原始图像的光流。

### 三、实验方法与步骤

#### 方法一：帧间差分法

运用传统帧差法提取运动目标的过程如下图 3-1 所示，首先，从图片序列中获取第  $k$  帧以及  $k-1$  帧进行平滑去噪，将其转换为灰度图像，然后通过帧差法得到二值化图像，最后再进行形态学去噪处理，得到实验结果。

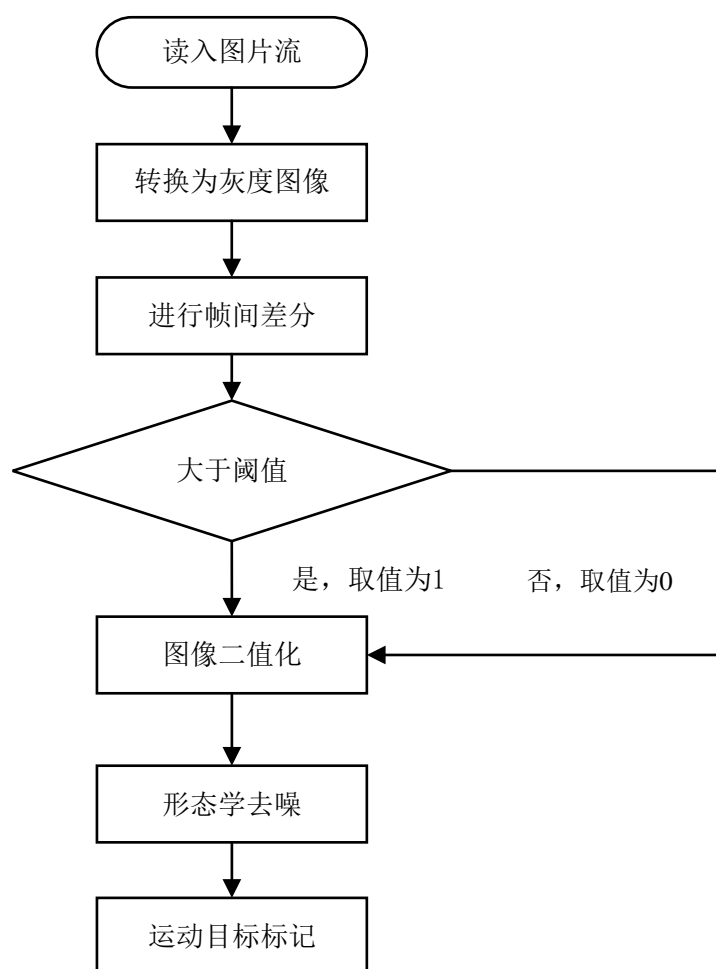


图 3-1 帧间差分运动目标检测流程图

主要步骤如下：

#### 1. 读入图片流，转化为灰度图像

以流的形式读取文件夹内的所有图片，保存在 `Cell_image` 中。在 `traffic` 文件中，一共含有连续的 9 帧图片。然后通过 `rgb2gray()` 函数把提取出来的彩色图像转化为灰度图像，这里也可采用公式，即

$$Y = 0.299 * R + 0.587 * G + 0.114 * B \quad (3-1)$$

将计算得出的  $Y$  值替换  $R$ 、 $G$ 、 $B$  的值，即得到所有帧的灰度图像。

## 2. 帧间差分

在该步骤中，我们将相邻两帧的灰度图像进行差分运算，即将第  $k$  帧图片减去第  $k-1$  帧图片。

实验要求给出一帧的分割结果，在这里，我们选择一帧图像作为参考图像。

## 3. 阈值化

在该步骤中，我们对上一步中得到的差分图像进行二值化。运算公式如下：

$$D(x, y) = \begin{cases} 255, & |f_k(x, y) - f_{k-1}(x, y)| \geq T \\ 0, & |f_k(x, y) - f_{k-1}(x, y)| < T \end{cases} \quad (T \text{ 为阈值}) \quad (3-2)$$

式中， $T$  为设定的阈值，当两帧之差大于阈值  $T$  时，差值取值为 255，当两帧之差绝对值小于阈值  $T$  时，差值取 0，经过这样的差分处理，我们就可以得到灰度图像的二值化图像。

阈值的选择非常关键。若阈值选择太大，则有可能造成检测出来的目标出现很多空洞，甚至有可能漏检；若阈值选择太小，那么将有可能出现大量的噪声，那么对于下一步的去噪处理将会产生很大的影响，因此，阈值的选择至关重要。

## 4. 形态学去噪，闭合操作

运用帧间差分法得到二值图像后，我们基本上就得到了运动目标的大致图像，但是根据运动目标检测的相关理论我们知道，运动的图像有许多干扰因素会产生噪声，从而影响检测效果。因此，得到二值图像后，再运用数学形态学的方法对二值图像进行去噪处理，从而可以得到更加准确的运动目标。

这里我们主要运用数学形态学的膨胀和腐蚀运算进行去噪处理。膨胀运算具有扩大图像的作用。将二值图像进行膨胀处理后，将会扩大图像的边缘，可以将检测出来的目标的边缘或者是内部的空洞化填充，从而在一定程度达到去除图像噪声的效果。而腐蚀运算与膨胀运算刚好相反，它对图像具有紧缩的作用，基于这种特性，我们运用腐蚀运算将所提取的目标的边缘多余的部分剔除掉，从而使目标轮廓更加清晰，使检测出来的目标更加精确。膨胀的运算公式为：

$$A \oplus B = \{x | A \cap Bx \neq \emptyset\} = \{x | Bx \uparrow A\} \quad (3-3)$$

由公式以及数学形态学基本理论我们可知，运用膨胀的方法，我们可以将二值图像中内部的空洞部分填充，或者将边缘缺失的部分补上，以确保图像的完整性。

与膨胀方法相反，腐蚀方法的运算公式为：

$$A \ominus B = \{x | B + x \subset A\} \quad (3-4)$$

由公式以及数学形态学的理论知识我们可以看出，腐蚀方法可以将二值图像中的边缘的“毛刺”剔除掉，把多余的部分锐化掉，使边缘轮廓更清晰，是原来处于一个整体的图像，有更清晰的纹路，从而得到更精确的运动目标。

## 5. 标记图像

得到经过形态学操作后的差分图像后，我们利用 `mask` 函数，根据选取的参考图像（像素值为 0 或 255）给原图像标记。

### 方法二：光流法

运用光流法提取运动目标的过程如下图 3-2 所示：

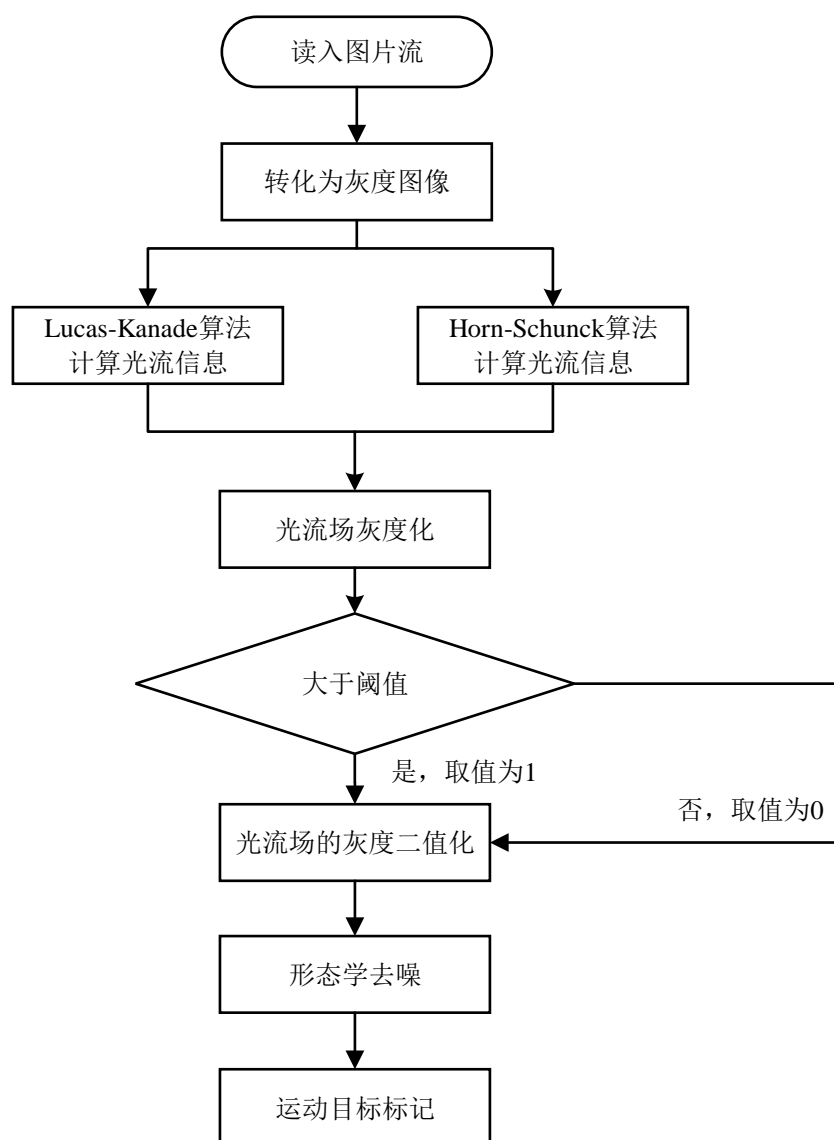


图 3-2 光流法运动目标检测流程图

主要步骤如下：

1. 读入图片流，转化为灰度图像。
2. 转化为灰度图像。
3. 计算光流信息，首先创建光流对象及类型转化对象。
4. 分别用 Horn-Schunck 算法和 Lucas-Kanade 算法调用光流对象计算两帧图片的光流。
5. 光流场灰度化。
6. 阈值化。  
在该步骤中，我们对上一步中得到的差分图像进行二值化。
7. 形态学去噪，闭合操作，先腐蚀后膨胀。  
得到二值图像后，再运用数学形态学的方法对二值图像进行去噪处理，从而可以得到更加准确的运动目标。
8. 标记图像  
得到经过形态学操作后的差分图像后，我们利用 mask 函数，根据选取的参考图像（像素值为 0 或 255）给原图像标记。



#### 四、实验结果及分析

帧间差分法实验结果分别如图 4-1 所示：

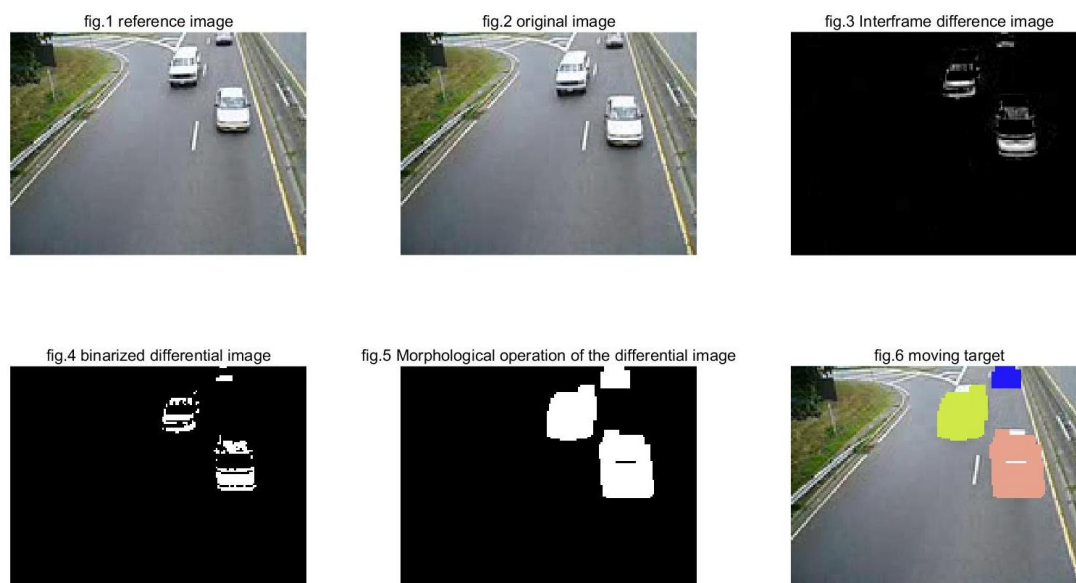


图 4-1 帧间差分法实验结果

Horn-Schunck 光流法、Lukas-Kanade 光流法实验结果分别如图 4-2、4-3 所示：

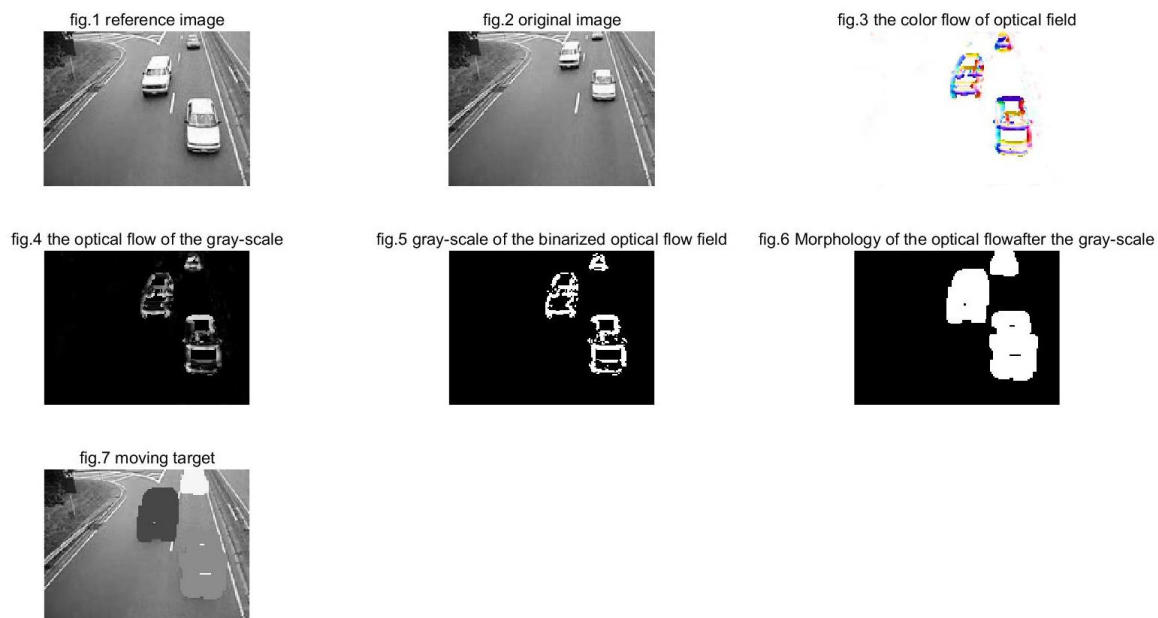


图 4-2 Horn-Schunck 光流法实验结果

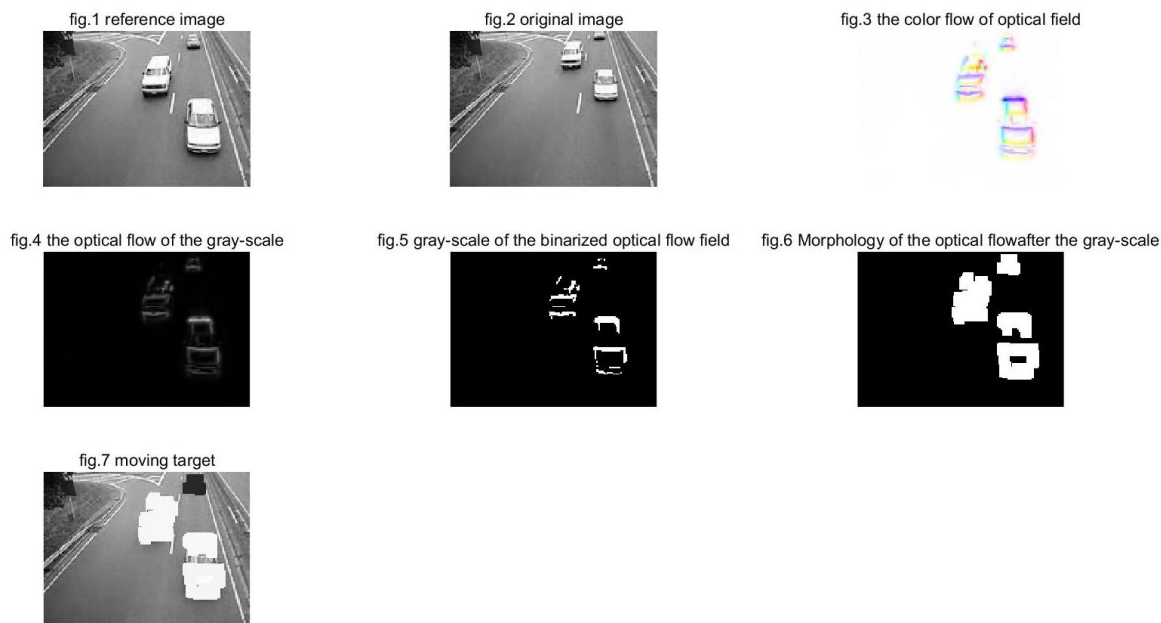


图 4-3 Lukas-Kanade 光流法实验结果

帧间差分法优缺点分析：

通常相邻两帧图像在时间上的间隔很短，所以当背景变化不是非常剧烈和迅速的时候可以利用前后两幅图像的差值来判断画面中运动的目标，通过设定阈值的调整可以改变方法检测的灵敏度。

优点：运算量小，实现简单，噪点较少，对光照不敏感。

缺点：不能检测速度过慢的物体，物体内部会造成空洞，物体本身也不完整。速度过快的物体，会形成两个前景区域。

光流法优缺点分析：

**Horn-Schunck** 光流法求得的是稠密光流，需要对每一个像素计算光流值，计算量比较大。而 **Lucas-Kanade** 光流法只需计算若干点的光流，是一种稀疏光流。

对画面中图像首先进行逐像素矢量赋值，当画面中出现运动目标时，运动目标在画面总所占据的像素点和背景像素点之间的矢量必然会有差异，如果没有运动目标出现则画面中的像素点的矢量变化应当为平滑的。正是由于有这样的差异，才能利用差异检测出运动目标在画面中的像素点。

优点：适合于背景不断变换的场景（移动摄像头），不需要预先的视频进行训练。

缺点：抗噪能力差，计算量大，光照影响比较大。

### 参考文献：

- [1] 章毓晋. 图像工程[M]. 清华大学出版社有限公司, 2005.
- [2] Lucas B D, Kanade T. An iterative image registration technique with an application to stereo vision[J]. 1981.
- [3] Shi J. Good features to track[C]//Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on. IEEE, 1994: 593-600.
- [4] Birchfield S. Derivation of kanade-lucas-tomasi tracking equation[J]. unpublished

notes, 1997.

[5] Bruhn A, Weickert J, Schnörr C. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods[J]. International Journal of Computer Vision, 2005, 61(3): 211-231.