计算机视觉——运动分析

2022年春季

桑农 王岳环

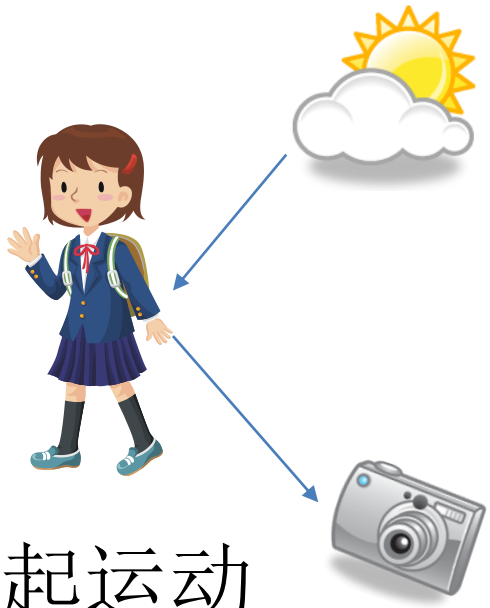
运动分析

- 序列图像和视频图像
  - 运动检测
  - 运动目标检测与定位
  - 运动目标分割和识别
  - 三维形状恢复
- 相关视觉应用
  - 目标跟踪
  - 背景建模
  - 行为分析（异常行为检测），场景理解

第11章 运动分析

- 11.1　运动分类和表达
- 11.2　全局运动检测
- 11.3　运动目标分割
- 11.4　运动光流和表面取向

运动起因



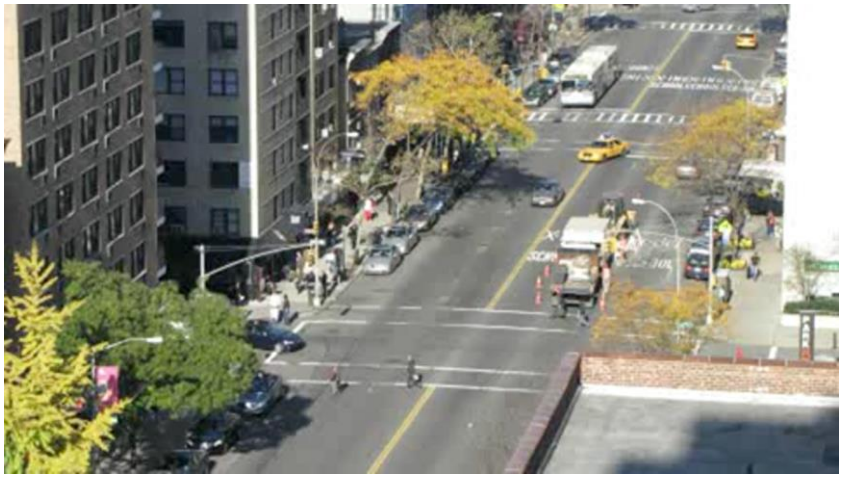
- 成像过程的三个因素
  - 光照
  - 物体
  - 相机
- 改变其中任意一个因素都会引起运动
  - 相机固定，物体运动（视频监控）
  - 相机移动，场景不动（3D恢复）
  - 相机移动，场景移动（体育比赛，电影）
  - 相机固定，目标运动，光源移动（延时摄影，time lapse）

运动举例

相机固定，目标移动

相机移动，场景不动

相机移动，场景移动



相机不动，目标移动，光源移动

运动举例

如何检测人手的运动信息

11.1 运动分类和表达

- 1. 运动分类
  - (1) 摄像机静止，景物运动
  - (2) 摄像机运动，景物静止
  - (3) 摄像机和景物都运动

  - (1) **前景运动**：前景运动指目标在场景中的自身运动，又称为局部运动
  - (2) **背景运动**：背景运动是由进行拍摄的摄像机的运动所造成的帧图像内所有点的整体移动，又称为**全局运动或摄像机运动**

11.1 运动分类和表达

摄像机的各种运动

- **跟踪**运动
- **升降**运动
- 进退或**推拉**运动
- **倾斜**运动
- **扫视**运动
- **滚转**运动
- **变焦**运动或缩放运动

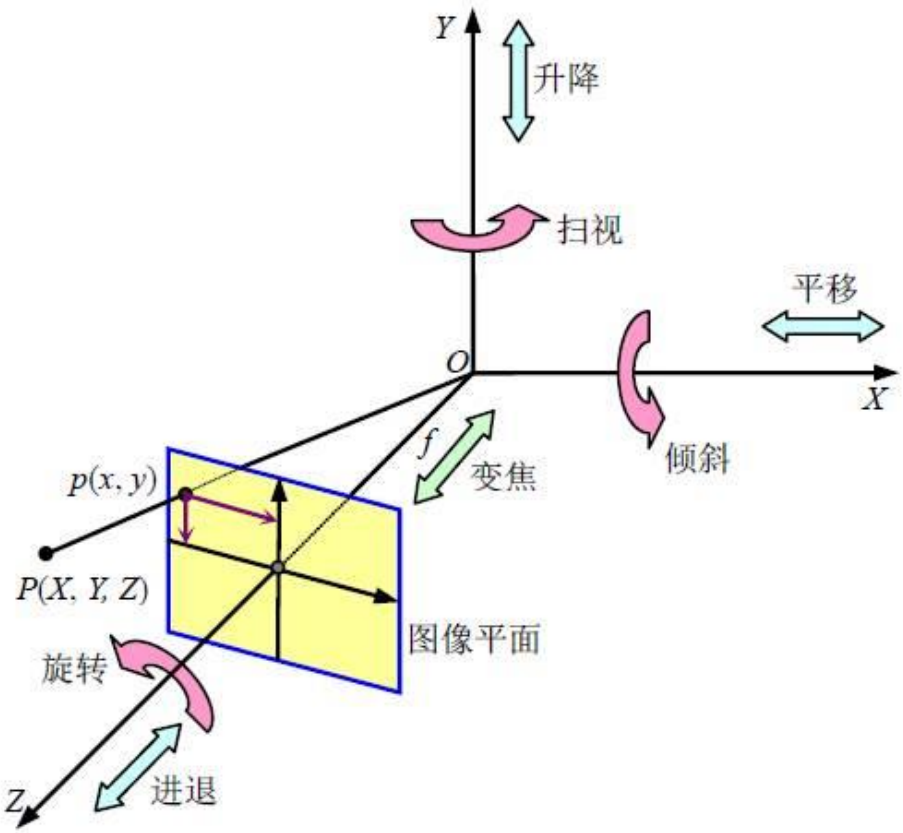


图 11.1.1 摄像机的各种运动

11.1 运动分类和表达

2. 运动矢量场表达

将每个**运动矢量**用（有起点）无箭头的线段（线段长度与矢量大小即运动速度成正比）来表示，并叠加在原始图像上

图 11.1.2 全局运动矢量叠加在原图上的结果

11.1 运动分类和表达

3. 运动直方图表达

（1）运动矢量方向直方图（紧凑表达）

- 仅保留运动的方向信息以减少数据量
- 需要考虑去除静止或基本静止点的影响

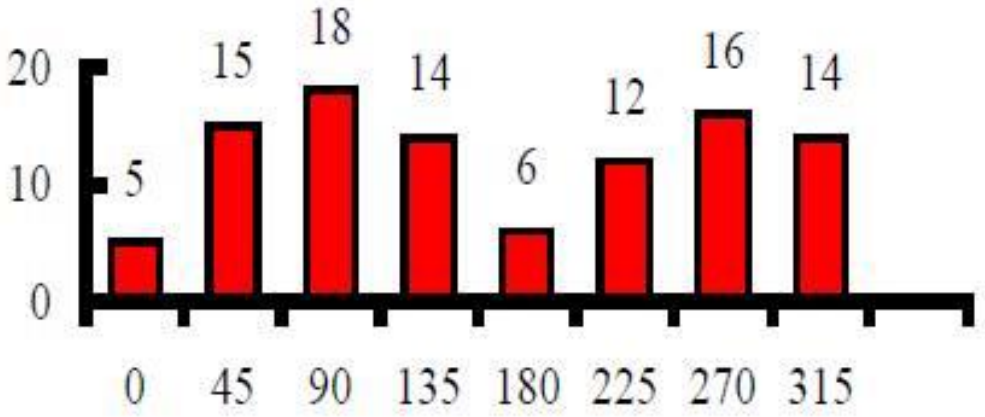


图 11.1.3 运动矢量方向直方图

11.1 运动分类和表达

3. 运动直方图表达

（2）运动区域类型直方图（紧凑表达）

- 运动类型，根据运动模型分类，即根据描述运动的参数模型（运动矢量）将运动分为不同的类型
- 仿射运动模型有6个参数，运动模型分类就是在6D的空间划分
- 可采用矢量量化的方法划分

11.1 运动分类和表达

3. 运动直方图表达

（2）运动区域类型直方图（紧凑表达）

借助对区域参数模型的表示来表达运动矢量场中各种运动的信息

步骤：

- 运动模型分类
- 将所有运动矢量量化到对应运动模型
- 统计每个运动类型对应的运动区域面积
- 将每个运动类型对应的面积组成直方图

11.1 运动分类和表达

3. 运动直方图表达

（2）运动区域类型直方图（紧凑表达）

借助对区域参数模型的表示来表达运动矢量场中各种运动的信息

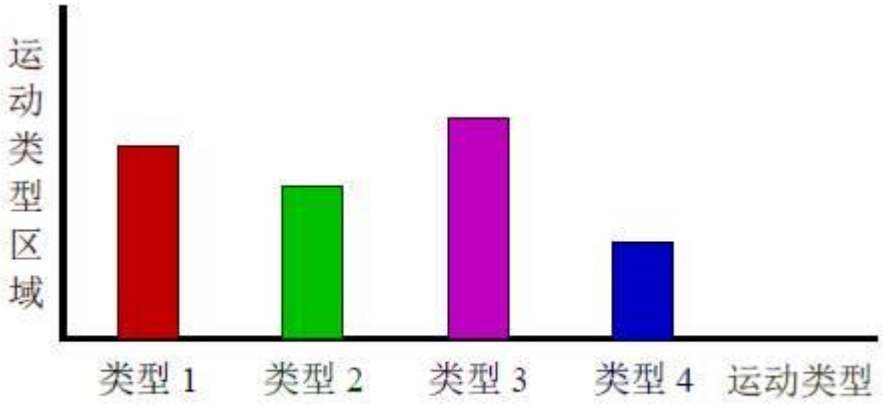
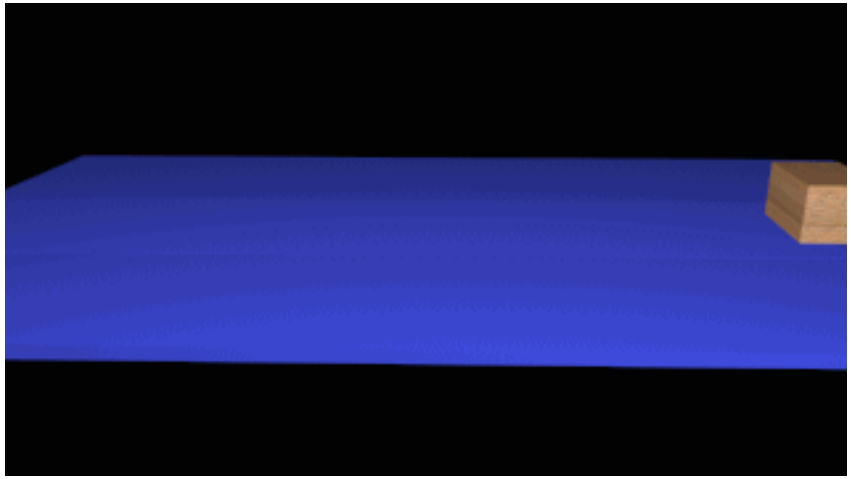


图 11.1.4 运动区域类型直方图

11.1 运动分类和表达

4. 运动轨迹表达

目标的**运动轨迹**表示了目标在运动过程中的位置信息



11.1 运动分类和表达

4. 运动轨迹表达

运动轨迹描述符由一系列关键点和一组在这些关键点间进行插值的函数构成

11.1 运动分类和表达

4. 运动轨迹表达

运动轨迹描述符由一系列关键点和一组在这些关键点间进行插值的函数构成（XYZ空间）

$$f(t) = f_p(t) + v_p(t - t_p) + \alpha_p(t - t_p)^2 / 2$$

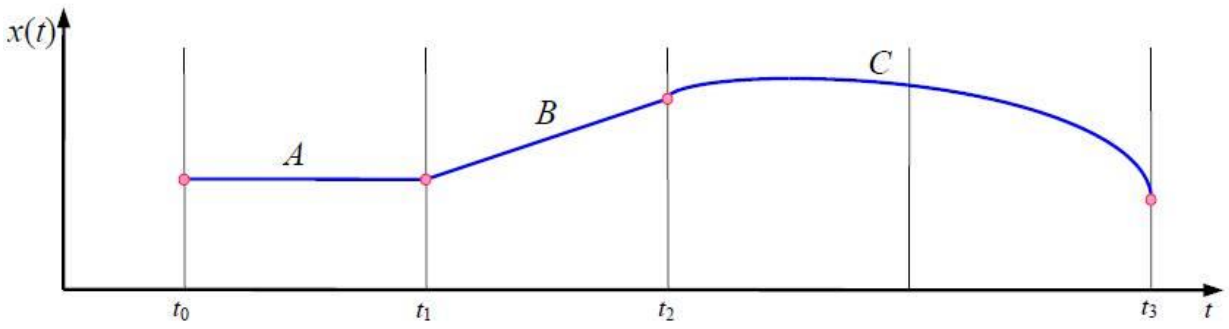


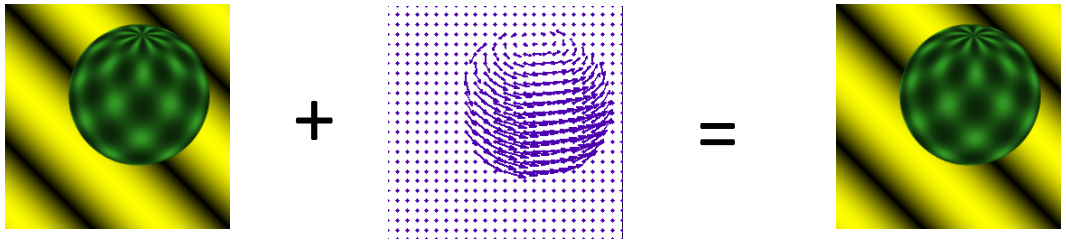
图 11.1.5 轨迹描述中关键点和插值函数的示意图

11.1.2 运动检测

- 视频分析需要检测感兴趣的运动信息
  - 1 基于**图像差**的运动检测

第一帧　　图像差　　第二帧

  - 2 基于**运动矢量**的运动检测



第一帧　　运动矢量　　第二帧

11.1.2 运动检测

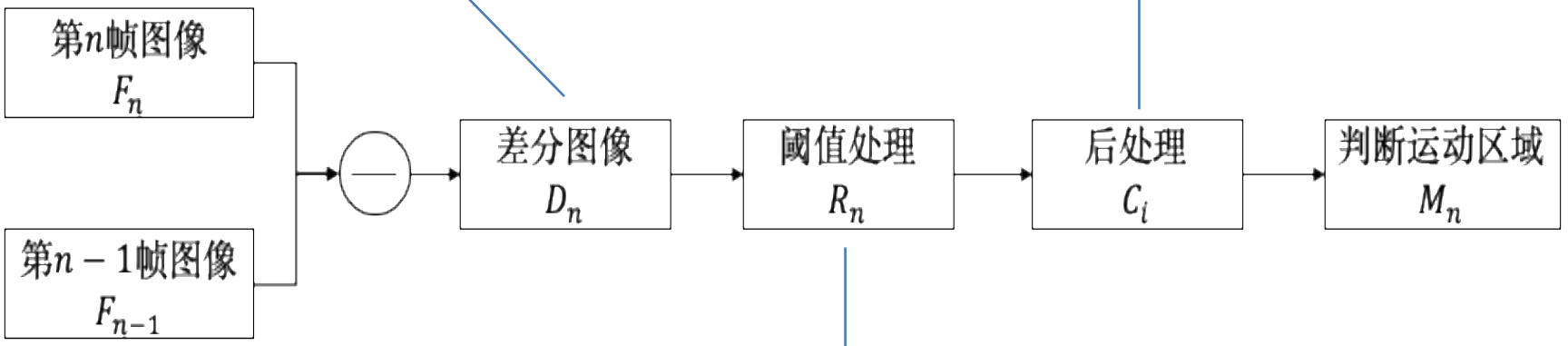
- 视频分析需要检测感兴趣的运动信息
  - 3 基于背景差分的运动检测

1. 基于图像差的运动检测

- 两帧帧间差分法

$$D_n(x,y) = |F_n(x,y) - F_{n-1}(x,y)|$$

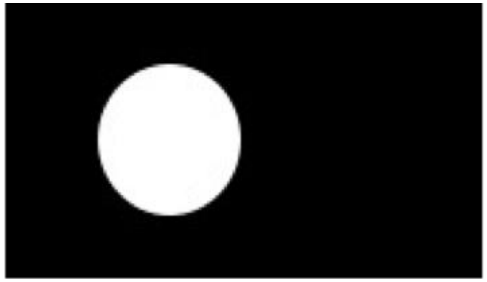
$$C_i = \begin{cases} 1 & A(C_i) > P \\ 0 & others \end{cases}$$



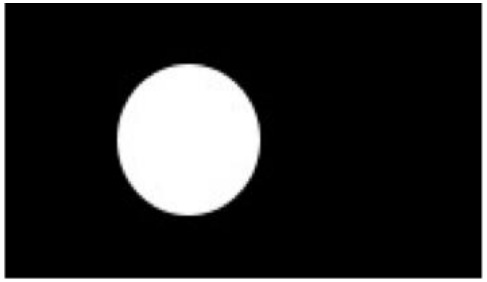
$$R_n(x,y) = \begin{cases} 1 & D_n(x,y) > T \\ 0 & others \end{cases}$$

1. 基于图像差的运动检测

- 两帧帧间差分法



(a) F_{n-1}



(b) F_n

(c) D_n

(d) R_n $(T = 10)$

(e) M_n $(P = 5)$

1. 基于图像差的运动检测

- 两帧帧间差分法
  - 两帧帧间差分法运动检测的优点为：
    - 算法实现简单，程序设计复杂度低，运行速度快；
    - 算法只依赖于短时间图像变化，动态环境自适应性强，对场景光线变化（非突变）不敏感。

1. 基于图像差的运动检测

- 两帧帧间差分法
  - 两帧帧间差分法运动检测的缺点为：
    - 不适用于相机运动（背景移动）的场景；
    - 会出现“空洞”现象
    - 会出现“双影”现象。可能会出现“两个”重叠的物体
    - 算法依赖于分割阈值，如果阈值选取过小则会引入噪声，如果阈值过大则可能会丢失部分运动区域。

1. 基于图像差的运动检测

1. 基于图像差的运动检测

- 基于相邻图像差的运动检测

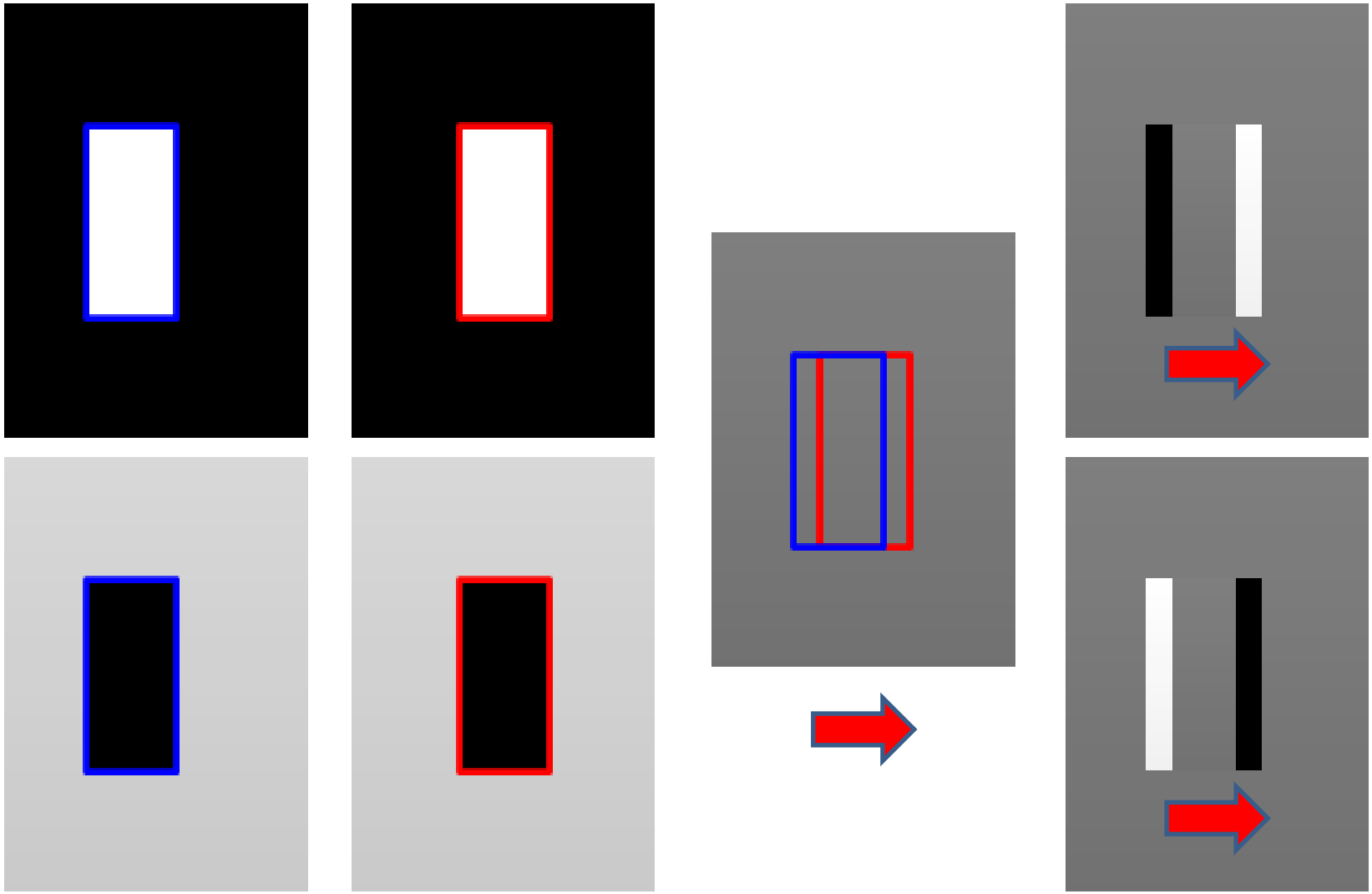
1. 基于图像差的运动检测

- 基于相邻图像差的运动检测

1. 基于图像差的运动检测

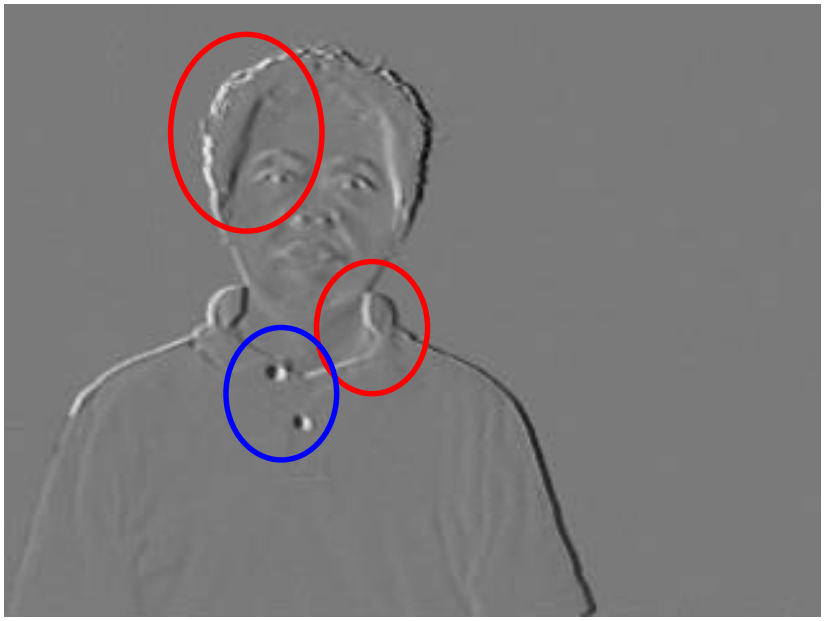
目标比背景亮

目标比背景暗



1. 基于图像差的运动检测

- 基于相邻图像差的运动检测



1. 基于图像差的运动检测

- 基于相邻图像差的运动检测

8帧图像之间的7个相邻图相差

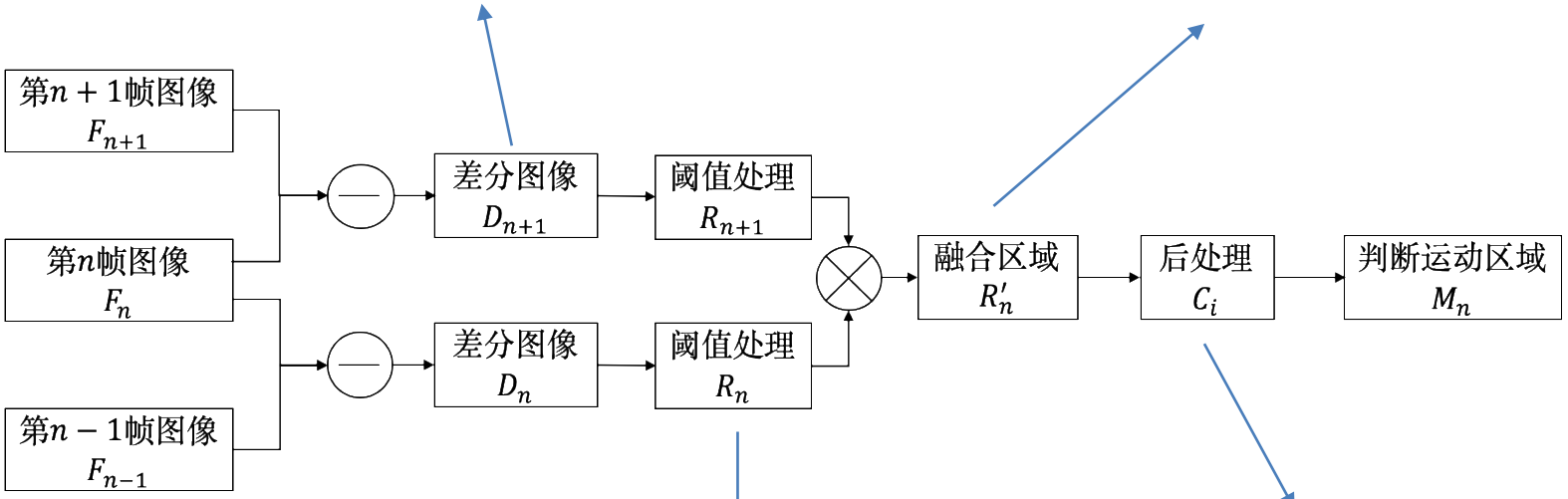
1. 基于图像差的运动检测

- 三帧帧间差分法

$$D_n(x,y) = |F_n(x,y) - F_{n-1}(x,y)|$$

$$D_{n+1}(x,y) = |F_{n+1}(x,y) - F_n(x,y)|$$

$$R'_n(x,y) = R_n(x,y) \cap R_{n+1}(x,y)$$



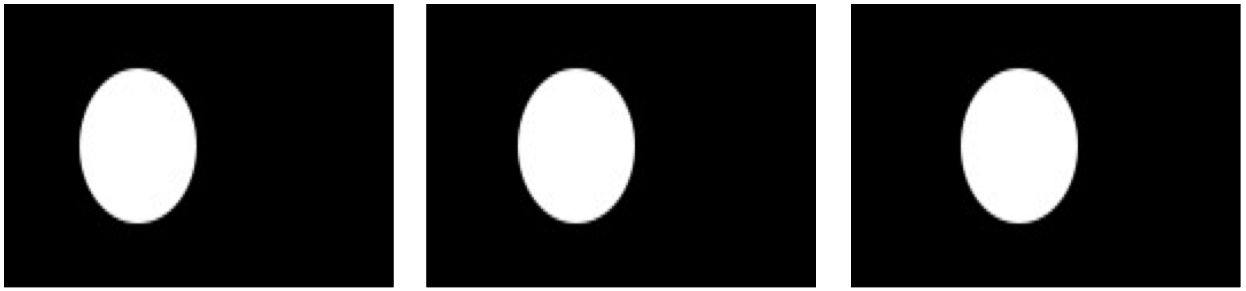
$$R_n(x,y) = D_n(x,y) > T$$

$$R_{n+1}(x,y) = D_{n+1}(x,y) > T$$

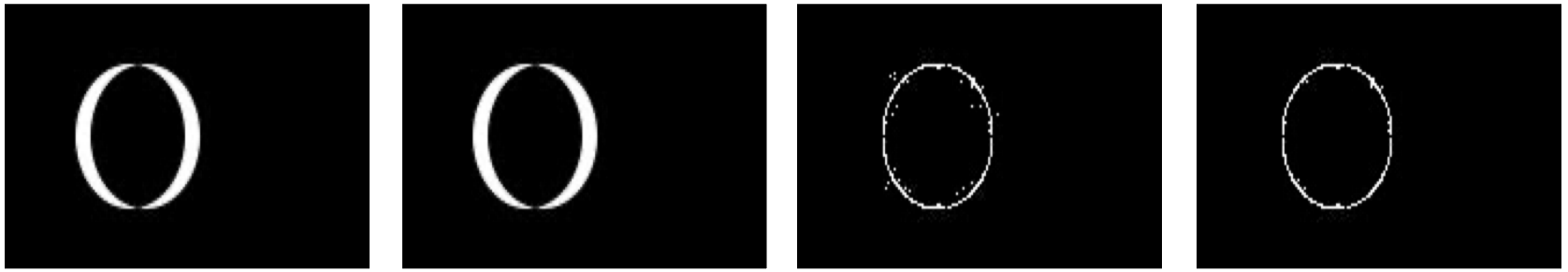
$$C_i = \begin{cases} 1 & A(C_i) > P \\ 0 & others \end{cases}$$

1. 基于图像差的运动检测

- 三帧帧间差分法



(a) F_{n-1} (b) F_n (c) F_{n+1}



(d) D_n (e) D_{n+1} (f) R'_n $(T = 10)$ (g) M_n $(P = 10)$

1. 基于图像差的运动检测

- 三帧帧间差分法
  - 三帧帧间差分法运动检测的优点为：
    - 算法实现简单，程序设计复杂度低，运行速度快；
    - 算法只依赖于短时间图像变化，动态环境自适应性强，对场景光线变化（非突变）不敏感；
    - 一定程度上避免了两帧帧间差分法的“双影”问题
  - 三帧帧间差分法运动检测的缺点为：
    - 不适用于相机运动（背景移动）的场景；
    - 会出现“空洞”现象。倾向于检测物体的边缘，容易造成运动区域的丢失或者不连续。
    - 算法依赖于分割阈值，如果阈值选取过小则会引入噪声，如果阈值过大则可能会丢失部分运动区域。

1. 基于图像差的运动检测

- 两帧帧间差分结果

(a) F_{n-1}

(b) F_n

(c) D_n

(d) R_n $(T = 10)$

(e) M_n $(P = 200)$

1. 基于图像差的运动检测

- 三帧帧间差分结果

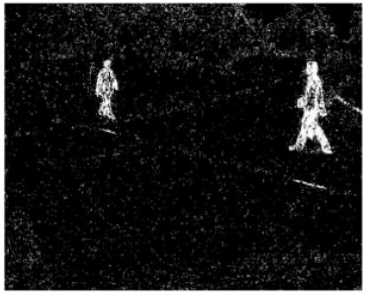
(a) F_{n-1}

(b) F_n

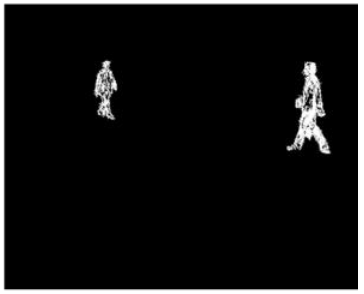
(c) F_{n+1}

(d) D_n

(e) D_{n+1}



(f) R'_n $(T = 10)$

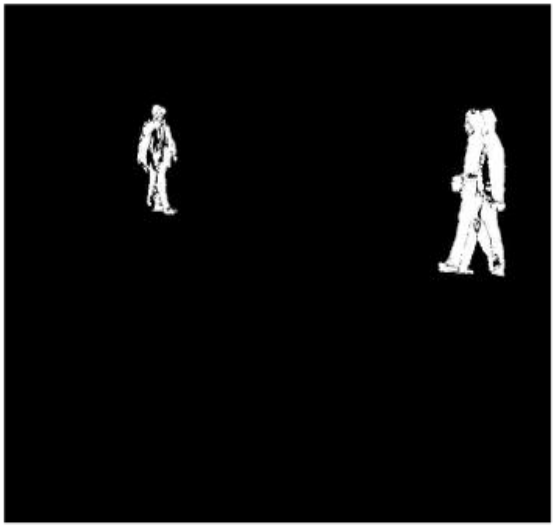


(g) M_n $(P = 200)$

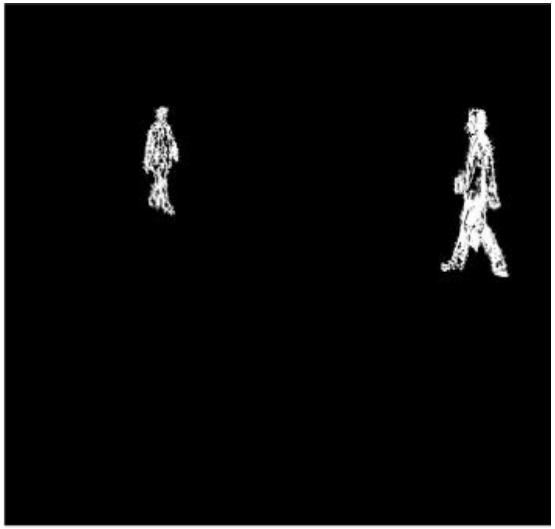
1. 基于图像差的运动检测

- 两帧帧间差分和三帧帧间差分结果

(a) 第n帧图像



(b) 两帧帧间差分结果



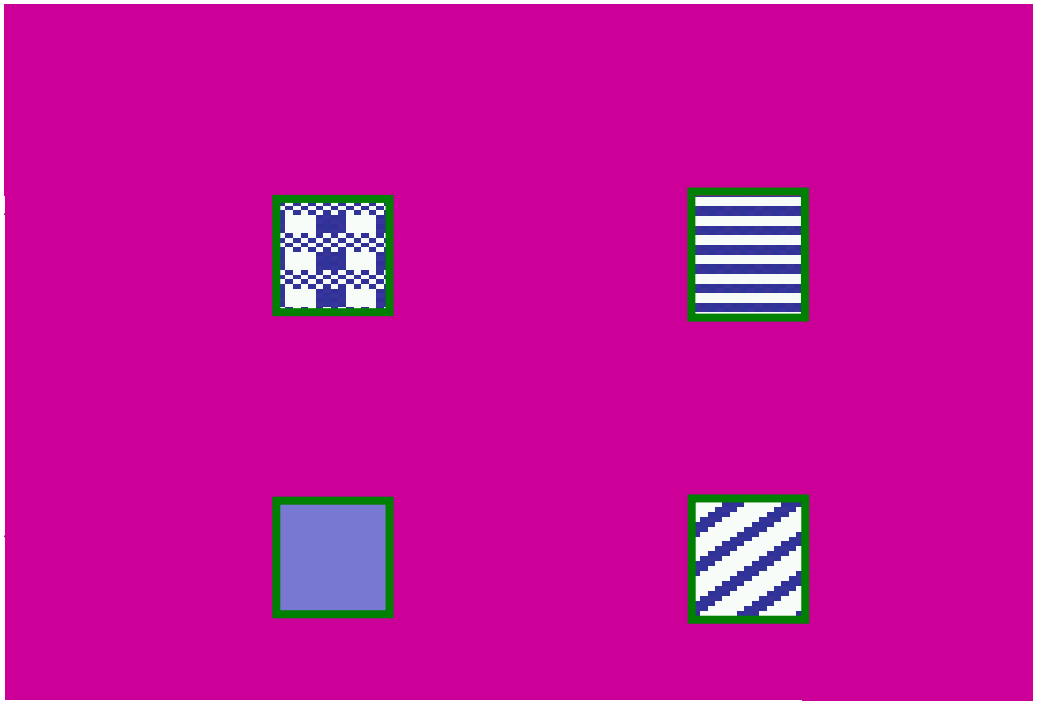
(c) 三帧帧间差分结果

如何检测运动目标

2. 基于运动矢量的运动检测

- 场景中不同目标以及背景的运动速度和方向一般不同：可以根据**运动矢量**来检测运动信息
- 基于运动矢量（光流）运动检测：
  - 计算运动矢量（光流）
  - 根据运动矢量（光流）进行分割

2. 基于运动矢量的运动检测



孔径问题

亮度模式不同
运动矢量不同

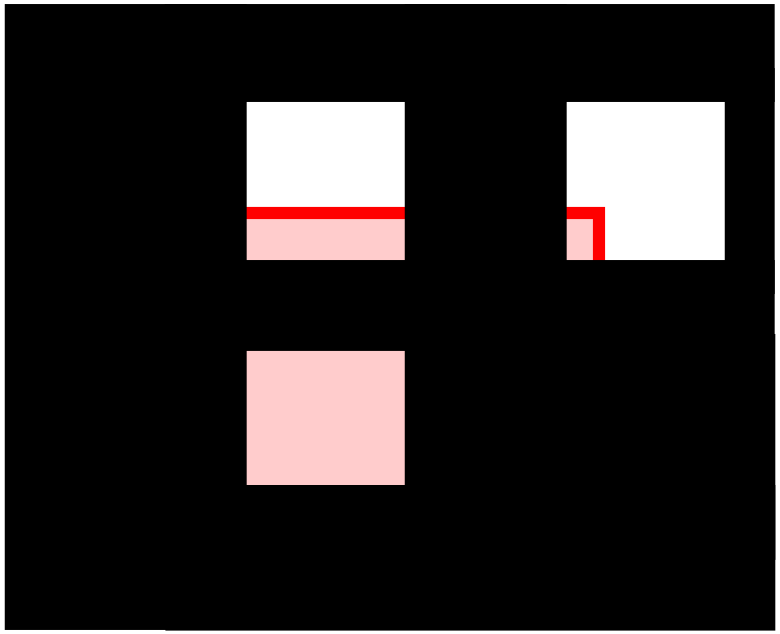
↓

运动矢量相同

↓

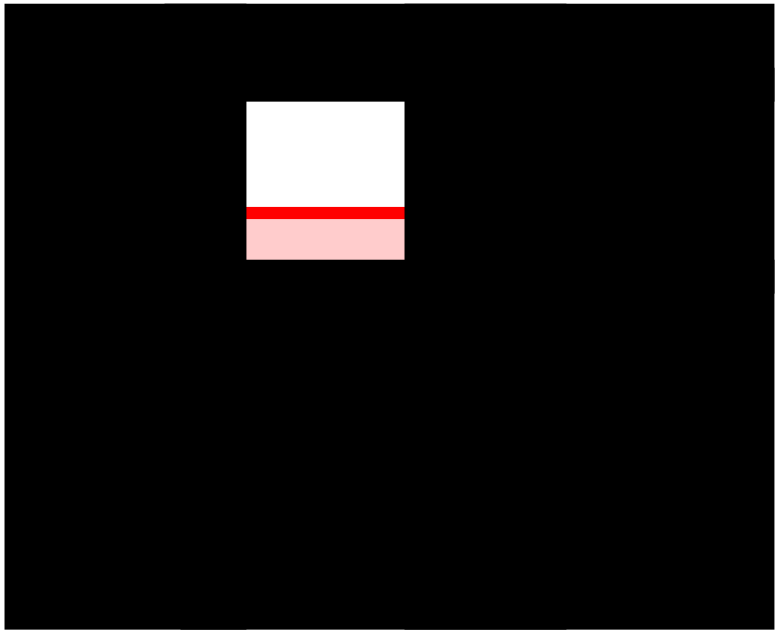
只有左上角窗口真实反映了运动的信息

2. 基于运动矢量的运动检测



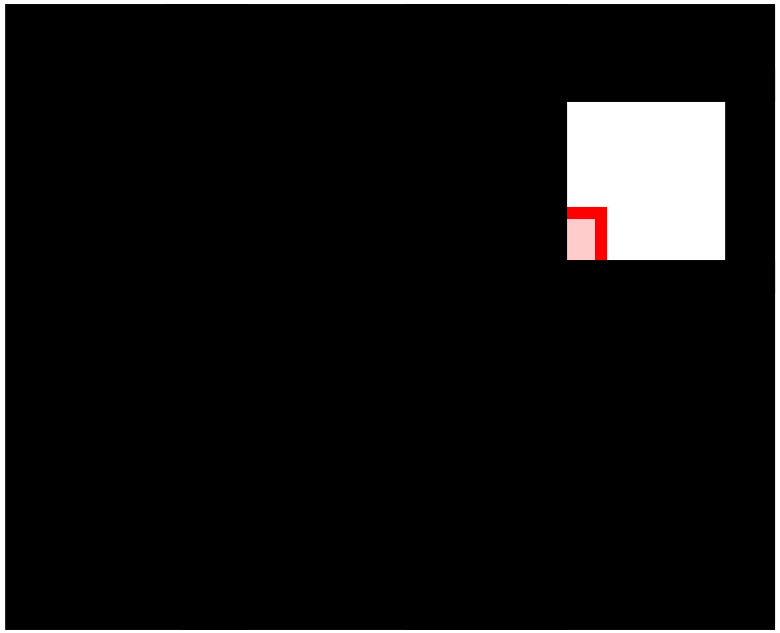
孔径问题

2. 基于运动矢量的运动检测



孔径问题

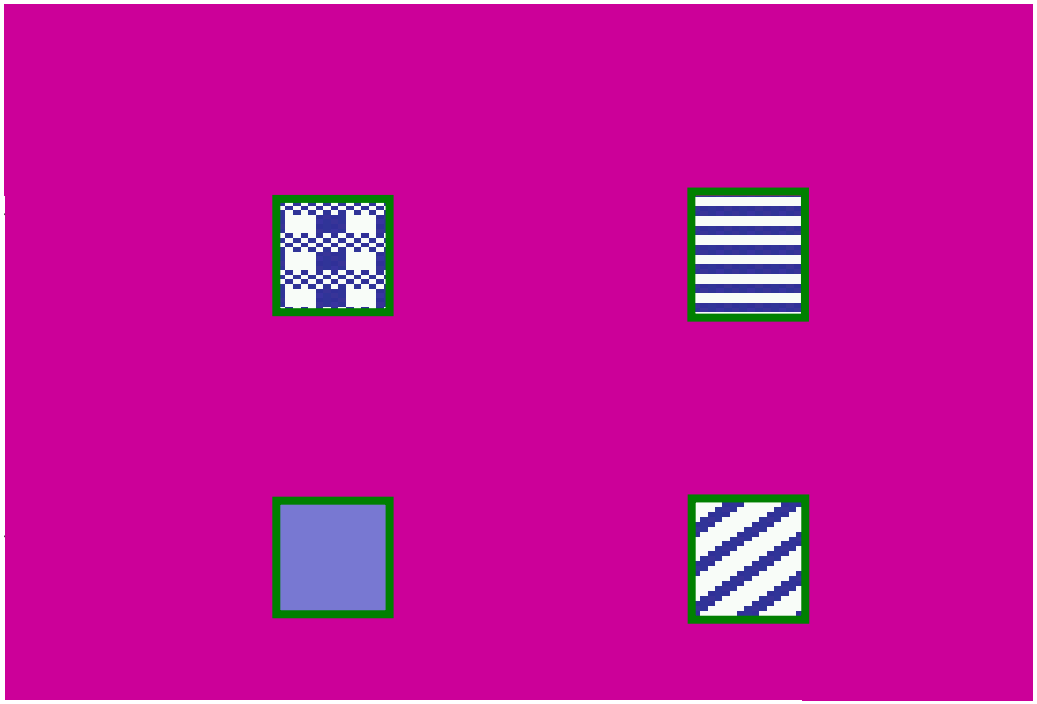
2. 基于运动矢量的运动检测



孔径问题

包括两条不同方向直线的窗口真实反映了运动信息

2. 基于运动矢量的运动检测



孔径问题

为什么少于两条方向不同的线时，不能准确计算运动矢量？

2.1 运动目标分割

- 从序列图像中检测运动目标，并将其分割出来
- 可用时域信息和空域信息

2.1 运动目标分割

1. 先分割之后再计算运动信息

直接利用时–空图像的灰度和梯度信息

（1）将视频帧分割成不同区域，对每个区域利用运动矢量场估计区域的仿射运动模型参数

优点：保留了区域边缘

缺点：会造成过分割

2.1 运动目标分割

1. 先分割之后再计算运动信息

直接利用时–空图像的灰度和梯度信息

（2）根据最小均方差准则将整个变化区域拟合到一个参数模型中，然后分成小区域逐次检测

2.1 运动目标分割

1. 先分割之后再计算运动信息

直接利用时–空图像的灰度和梯度信息

2. 先计算运动信息再分割

先估计光流场，然后基于光流场进行分割

- 在运动矢量场基础上进行分割可以保证运动边界响应较大
- 避免过分割

2.1 运动目标分割

1. 先分割之后再计算运动信息

直接利用时–空图像的灰度和梯度信息

2. 先计算运动信息再分割

先估计光流场，然后基于光流场进行分割

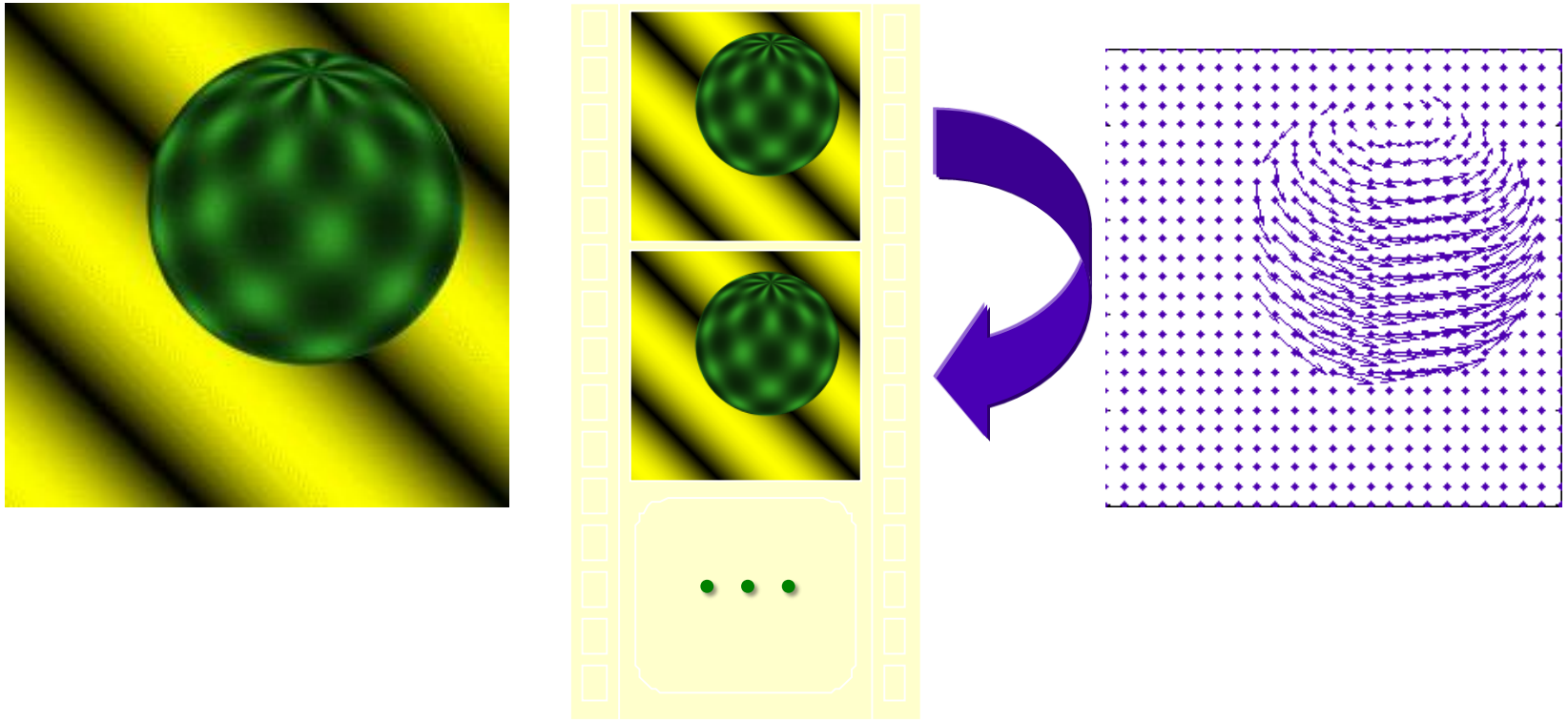
3. 同时计算运动信息和进行分割

同时采用两种方法，利用马尔可夫随机场及最大后验概率框架相联系，一般需要相当大的计算量

2.2 运动光流

- 当摄像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为**光流**，或者说物体带光学特征的部位的移动投影到视网膜平面（即图像平面）上就形成光流
- 光流有三个要素：一是运动（速度场），这是光流形成的必要条件；二是带光学特性的部位（例如有灰度的像素点），它能携带信息；三是成像投影 （从场景到图像平面），因而能被观察到

2.2 运动光流

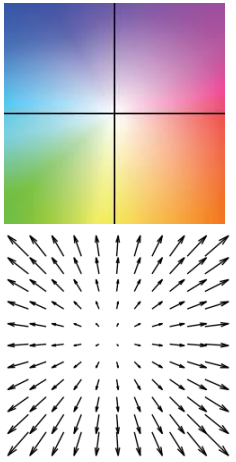


2.2 运动光流

- 可视化：密集显示
  - 将光流向量映射到颜色空间
  - 幅值: 饱和度　　方向: 色度

输入两幅图像

光流场

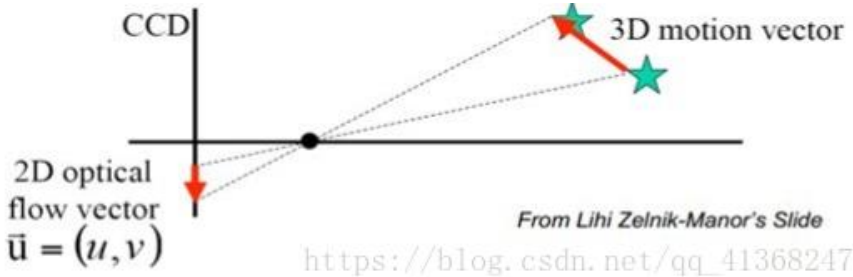


可视化码本

2.2 运动光流

- **光流基本原理**
  - 基于模型的运动检测不能描述任意视频中的运动
  - 光流场：设(x,y)处的流向量$(\mu(x,y),v(x,y))$

    $\mu=dx/dt$　　$v=dy/dt$



- **亮度恒常性假设**

2.2 运动光流

时刻$t + \mathrm{d}\,t$在图像点$(x + \mathrm{d}\,x, y + \mathrm{d}\,y)$处的照度应当与时刻$t$在图像点$(x, y)$的照度相同

$$f(x, y, t) = f(x + \mathrm{d}x,\ y + \mathrm{d}y,\ t + \mathrm{d}t)$$

用泰勒级数展开，令$\mathrm{d}t \to 0$，取极限并略去高阶项

$$-\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial f}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} = \nabla \boldsymbol{f} \bullet \boldsymbol{w}$$

$\omega = (u, v)$

光流约束方程: 灰度的（一阶）时间变化率是场景亮度变化率与该点运动速度的乘积

$$(E_x,\ E_y) \bullet (u,\ v) = -E_t$$

2.2 运动光流

时刻$t + \mathrm{d}\,t$在图像点$(x + \mathrm{d}\,x, y + \mathrm{d}\,y)$处的照度应当与时刻$t$在图像点$(x, y)$的照度相同

$$f(x, y, t) = f(x + \mathrm{d}x,\ y + \mathrm{d}y,\ t + \mathrm{d}t)$$

光流约束方程：

$$-\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial f}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} = \nabla \boldsymbol{f} \bullet \boldsymbol{w}$$

也可以通过全微分得到：

$$E_x = \partial f/\partial x \quad E_y = \partial f/\partial y \quad E_t = \partial f/\partial t$$

$$E_x u + E_y v + E_t = 0$$

$$(E_x,\ E_y) \bullet (u,\ v) = -E_t$$

2.2 运动光流

光流计算指对光流约束方程求解，即根据图像点灰度值的梯度求光流分量

光流约束方程是关于速度分量u,v的一条直线

仅一个光流约束方程不足以唯一确定u和v

$$u_0 = -E_t / E_x \qquad v_0 = -E_t / E_y \qquad \theta = \arctan(E_x / E_y)$$

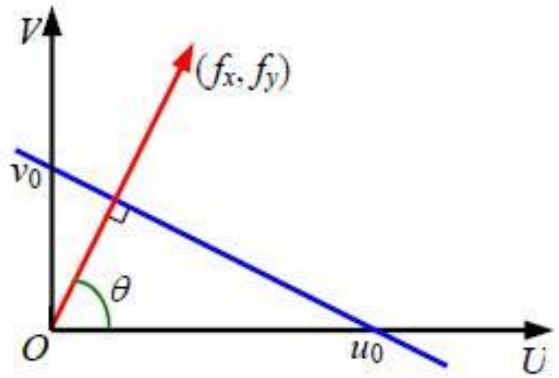


图 11.4.1 满足光流约束方程的 u 和 v 值在一条直线上

2.2 运动光流

1. Lucas-Kanade光流法（空间一致性假设）

假设图像上相邻点的速度也是一致的，即图像中P点附近点的邻域位移（或速度变化）相同

$$\begin{bmatrix} E_x(P_1) & E_y(P_1) \\ E_x(P_2) & E_y(P_2) \\ \vdots & \vdots \\ E_x(P_n) & E_y(P_n) \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} E_t(P_1) \\ E_t(P_2) \\ \vdots \\ E_t(P_n) \end{bmatrix}$$

2.2 运动光流

2. Horn-Schunck光流法（全局平滑假设）

速度的空间变化率为零

$$(\nabla u)^2 = \left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\right)^2 = 0 \quad (\nabla v)^2 = \left(\frac{\partial v}{\partial x} + \frac{\partial v}{\partial y}\right)^2 = 0$$

将这两个条件与光流约束结合，最小化：

$$\varepsilon = \sum_x \sum_y \left\{ (E_x u + E_y v + E_t)^2 + \lambda^2 \left[(\nabla u)^2 + (\nabla v)^2 \right] \right\}$$

光流约束方程

平滑约束

2.2 运动光流

1. Horn-Schunck光流法（全局平滑假设）

$$\varepsilon = \sum_x \sum_y \left\{ (E_x u + E_y v + E_t)^2 + \lambda^2 \left[(\nabla u)^2 + (\nabla v)^2 \right] \right\}$$

将 ε 对u和v分别求导并取导数为零（Euler方程）

$$E_x^2 u + E_x E_y v = -\lambda^2 \nabla u - E_x E_t$$

$$E_x^2 v + E_x E_y u = -\lambda^2 \nabla v - E_y E_t$$

图像中：令$\bar{u}$ $\bar{v}$ 表示u邻域和v邻域中的均值

$$\nabla u = u - \bar{u} \quad \nabla v = v - \bar{v}$$

得到：

$$(E_x^2 + \lambda^2) u + E_x E_y v = \lambda^2 \bar{u} - E_x E_t$$

$$(E_y^2 + \lambda^2) v + E_x E_y u = \lambda^2 \bar{v} - E_y E_t$$

2.2 运动光流

1. Horn-Schunck光流法（全局平滑假设）

$$u = \bar{u} - \frac{E_x(E_x\bar{u} + E_y\bar{v} + E_t)}{\lambda^2 + E_x^2 + E_y^2}$$

$$v = \bar{v} - \frac{E_y(E_x\bar{u} + E_y\bar{v} + E_t)}{\lambda^2 + E_x^2 + E_y^2}$$

松弛迭代方程

$$u^{(n+1)} = \bar{u}(n) - \frac{E_x[E_x\bar{u}(n) + E_y\bar{v}(n) + E_t]}{\lambda^2 + E_x^2 + E_y^2}$$

$$v^{(n+1)} = \bar{v}(n) - \frac{E_y[E_x\bar{u}(n) + E_y\bar{v}(n) + E_t]}{\lambda^2 + E_x^2 + E_y^2}$$

2.2 运动光流

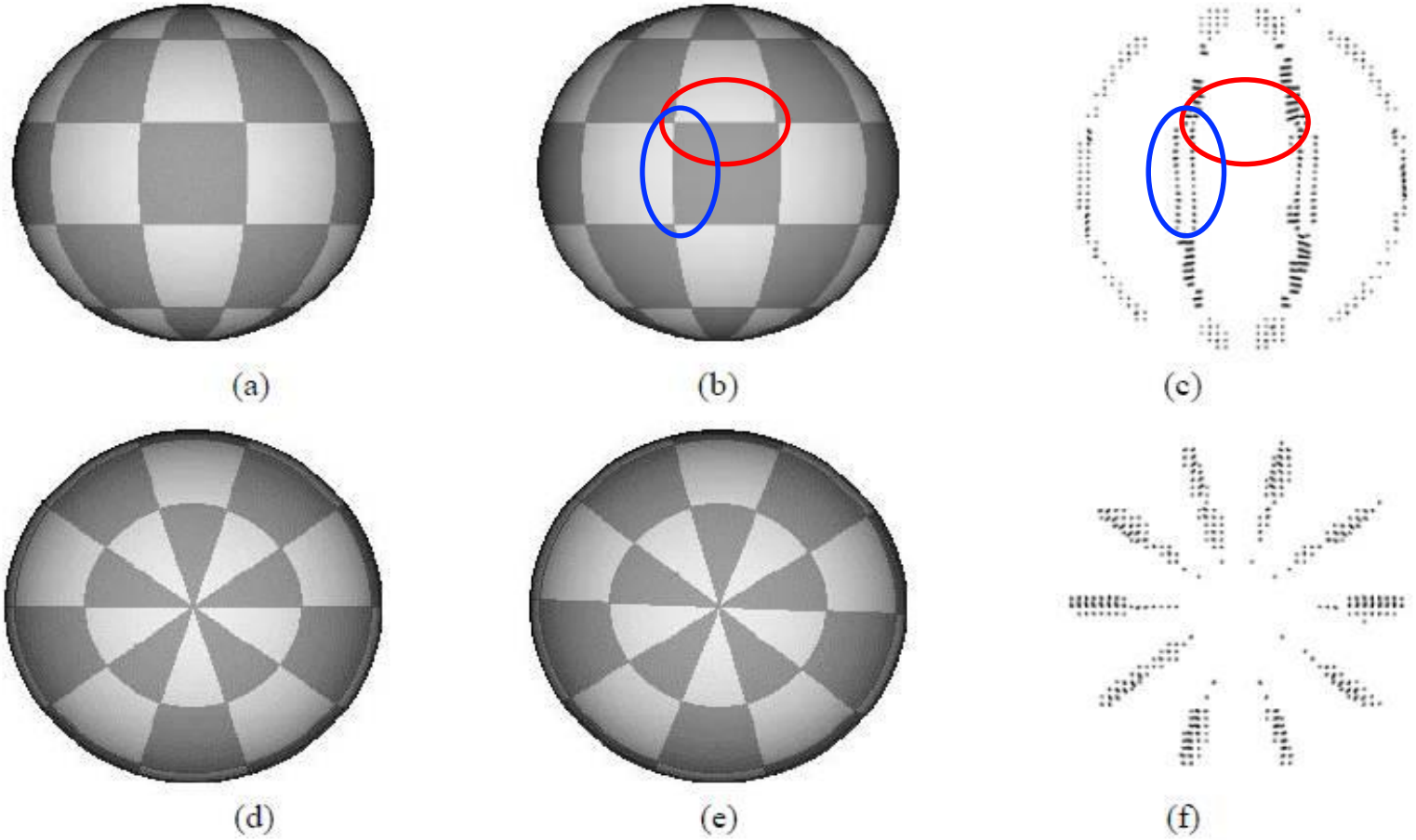


图 11.4.3 光流检测示例

2.2 运动光流

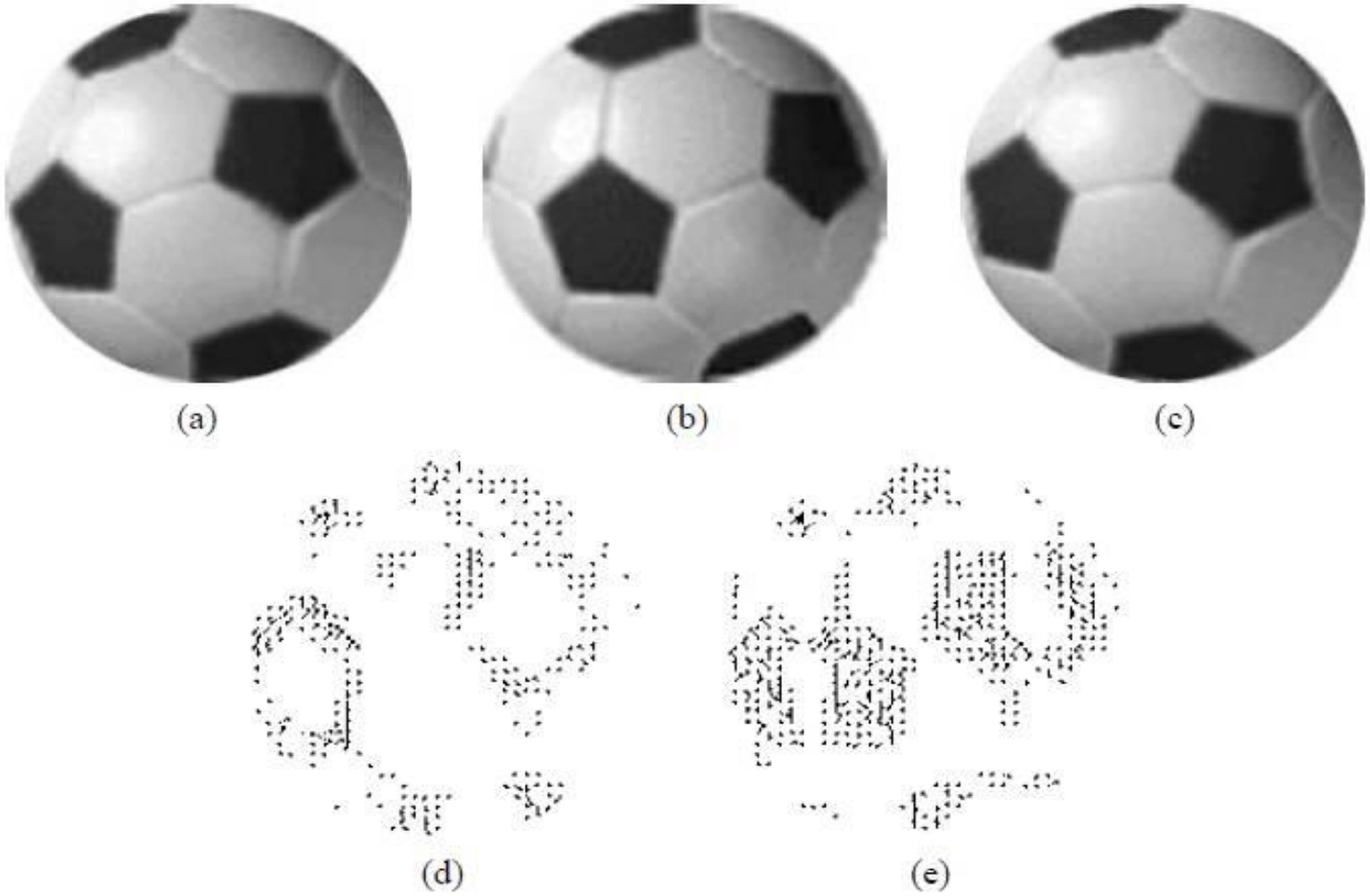


图 11.4.4 光流检测实例

2.2 运动光流

光流检测结果

(a) 第$n-1$帧图像

(b) 第n帧图像

(c) Lucas-Kanade光流结果

(d) Horn-Schunck光流结果

2.2 运动光流

- 优点
  - 光流法是基于像素点的亮度信息计算的，不需要事先知道场景信息就可以计算场景中运动目标的信息；
  - 光流法可以应用于背景静止的应用场景，也可以应用于背景运动的场景，根据前景目标和背景的运动不同可以区分感兴趣的运动物体；
  - 光流不仅携带了运动物体的信息，而且还携带了物体三维信息，可以根据物体的光流估计物体的形状，即基于运动的形状分析。

2.2 光流法优缺点

- 缺点
  - 光流法必须满足两个基本假设，这在实际中往往不能满足
  - 孔径问题。孔径问题的本质是欠约束，不能从单个采样中得到唯一解
  - 光流法的准确性和效率难以兼顾。稠密光流通常有较好的精度，但是其计算量大、耗时长，不适合实时视频处理的情况。而稀疏光流可以极大提升光流计算的速度，但是其精度和准确性又难以保证。

The end !