



第十二章：运动分析

中国科学技术大学
电子工程与信息科学系

主讲教师：李厚强 (lihq@ustc.edu.cn)

周文罡 (zhwg@ustc.edu.cn)

李 礼 (lil1@ustc.edu.cn)

胡 洋 (eeyhu@ustc.edu.cn)

助教：王 宁 (wn6149@mail.ustc.edu.cn)

席 茂 (ximao@mail.ustc.edu.cn)

姚 鑫 (xinyao@mail.ustc.edu.cn)

赵 鉴 (zj140@mail.ustc.edu.cn)



运动分析

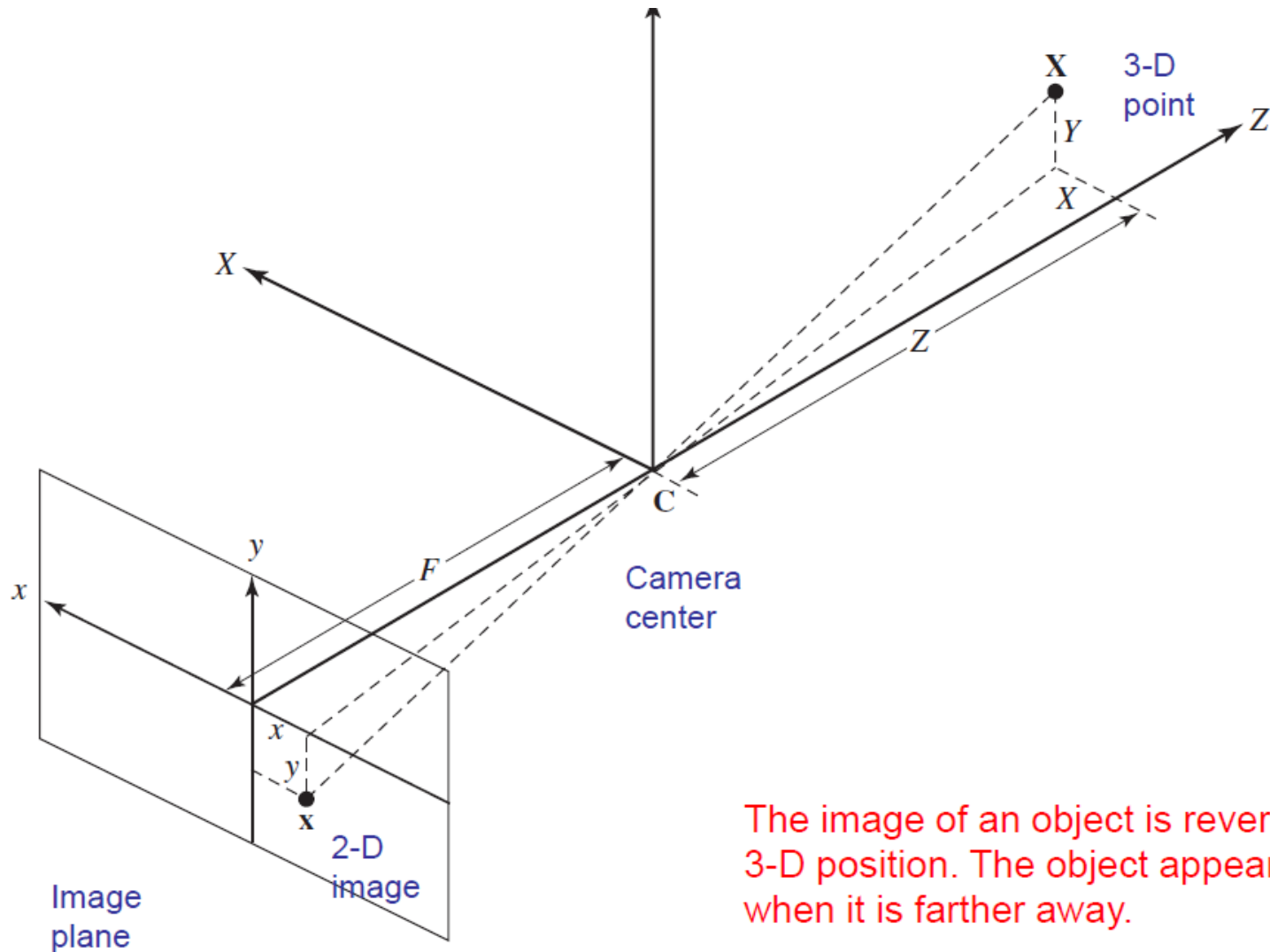
- **二维运动模型**
- 二维运动 vs. 光流
- 运动估计中的一般方法
- 基于像素的运动估计
- 基于块的运动估计
- 多分辨率运动估计
- 相位相关法
- 可形变块匹配
- 全局运动估计
- 局部运动估计
- 运动分割



二维运动模型

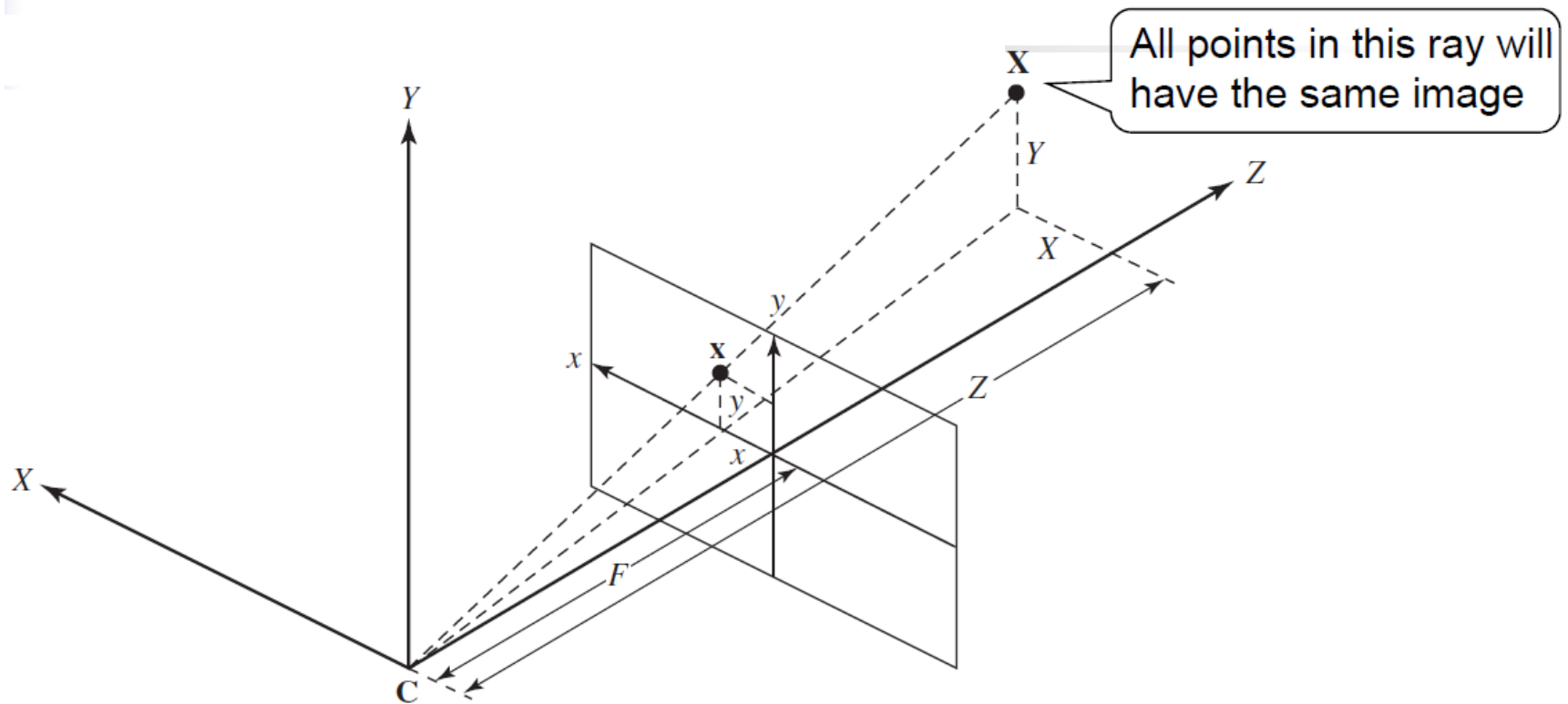
- 相机投影
- 三维运动
- 三维运动的投影
- 刚体目标的二维运动
 - 投影映射
- 投影映射的近似
 - 仿射模型
 - 双线性模型

针孔相机



The image of an object is reversed from its 3-D position. The object appears smaller when it is farther away.

针孔相机模型：透视投影

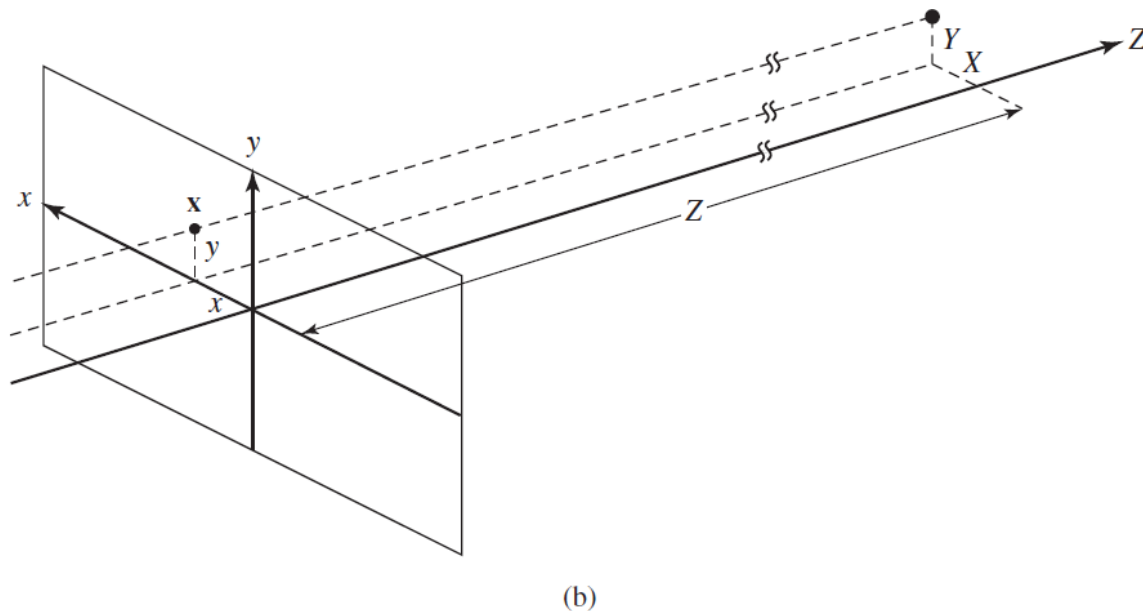


(a)

$$\frac{x}{F} = \frac{X}{Z}, \frac{y}{F} = \frac{Y}{Z} \Rightarrow x = F \frac{X}{Z}, y = F \frac{Y}{Z}$$

x, y are inversely related to Z

近似模型: 正交投影



当一个目标很远时 ($Z \rightarrow \infty$), $x = X, y = Y$ 。

只要物体表面相对深度的变化与物体到相机的距离相比可以忽略时, 就可以使用这个近似。



三维运动——刚体运动模型

刚体运动模型： $X' = R \cdot X + T$

$X' = X + T$ 平移

旋转 $[R] = [R_z] \cdot [R_y] \cdot [R_x]$

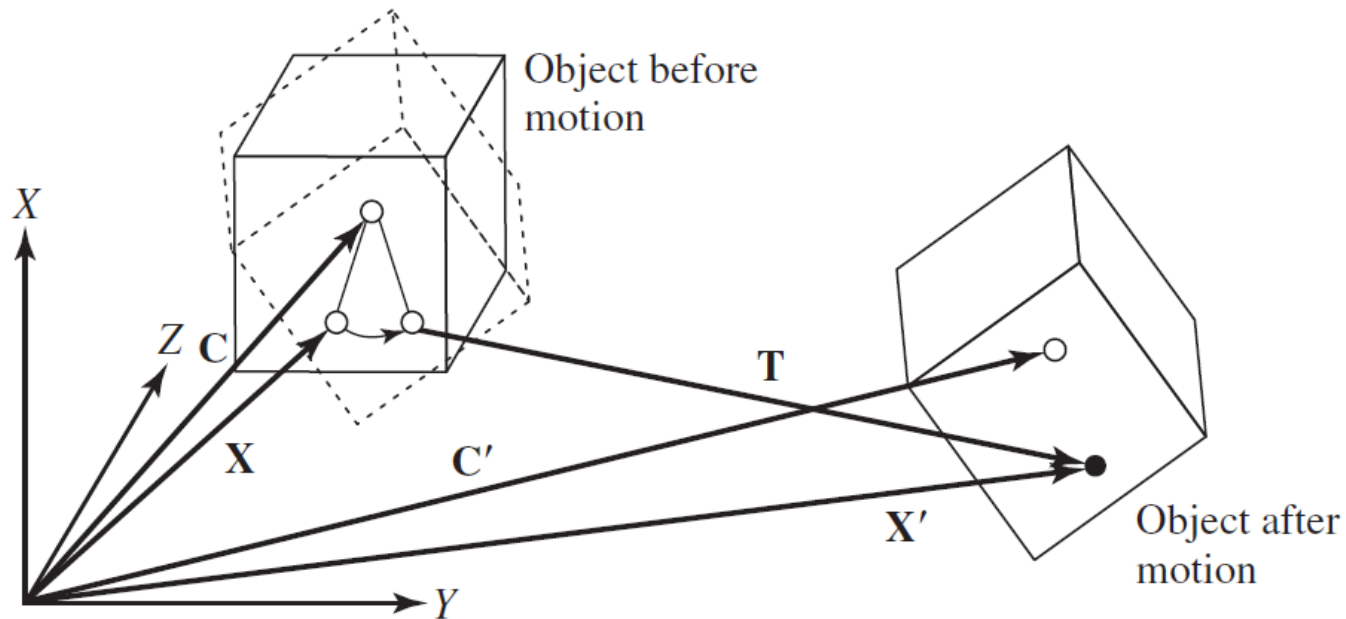
$$[R_x] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{bmatrix}$$

$$[R_y] = \begin{bmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{bmatrix}$$

$$[R_z] = \begin{bmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[R] \approx [R'] = \begin{bmatrix} 1 & -\theta_z & \theta_y \\ \theta_z & 1 & -\theta_x \\ -\theta_y & \theta_x & 1 \end{bmatrix}$$

刚性物体运动



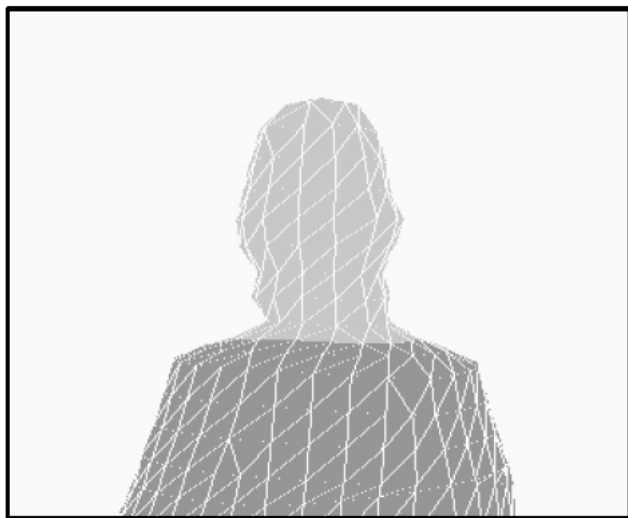
Rotation and translation wrt. the object center :

$$\mathbf{X}' = [\mathbf{R}] (\mathbf{X} - \mathbf{C}) + \mathbf{T} + \mathbf{C}; \quad [\mathbf{R}] : \theta_x, \theta_y, \theta_z; \quad \mathbf{T} : T_x, T_y, T_z$$

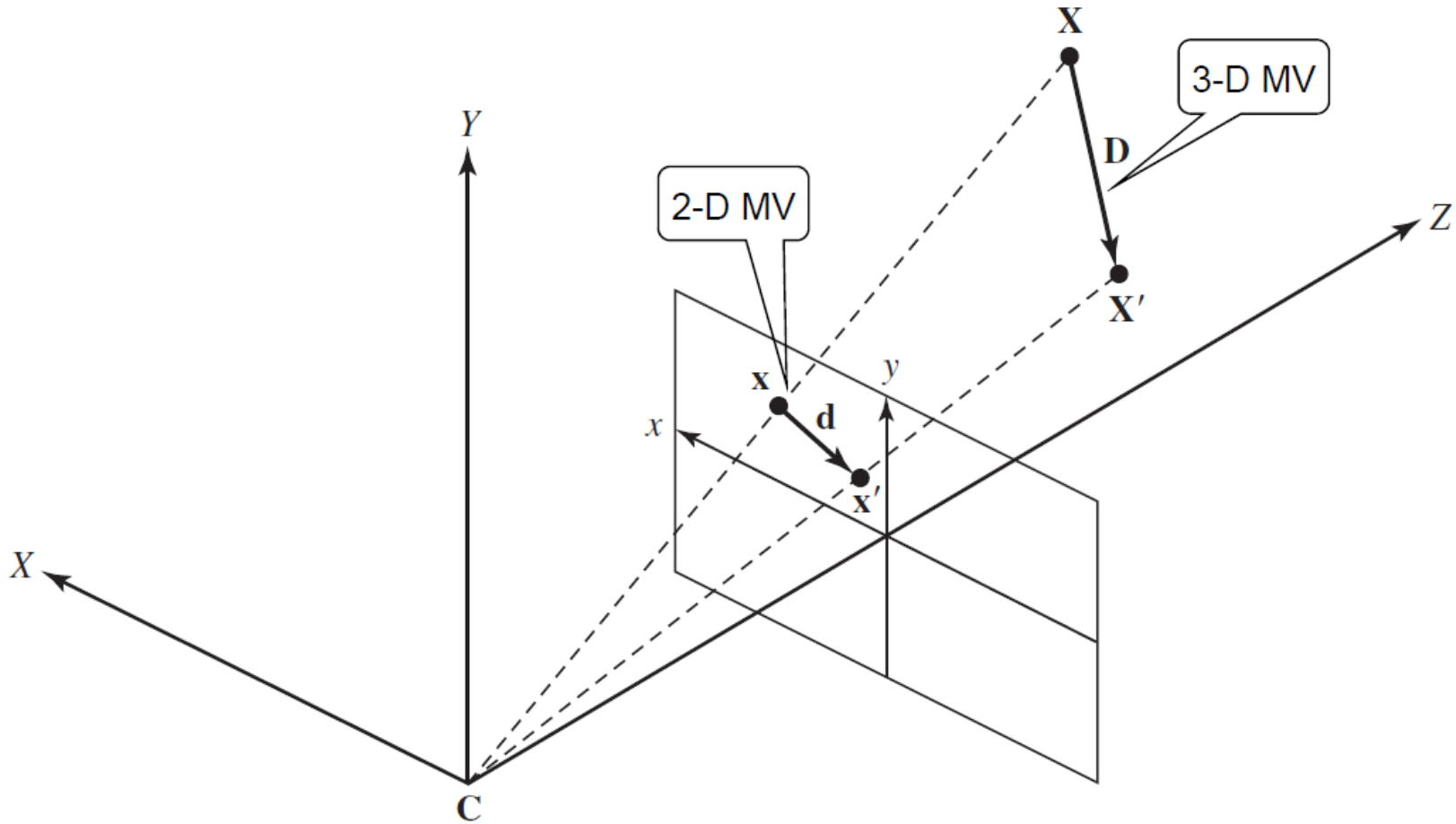
柔性物体运动

□ 两种描述方式

- 分解为多个互相连接的刚性子物体
- 全局运动加子目标的局部运动
 - ✓ 例如：人体由多个肢节组成，每个肢节在做刚体运动



三维与二维运动之间的关系



MV: motion vector



定义和符号

□ 三维运动向量

$$D(X; t_1, t_2) = X' - X = [D_X, D_Y, D_Z]^T$$

□ 二维运动向量

$$d(X; t_1, t_2) = X' - X = [d_x, d_y]^T$$

□ 映射函数

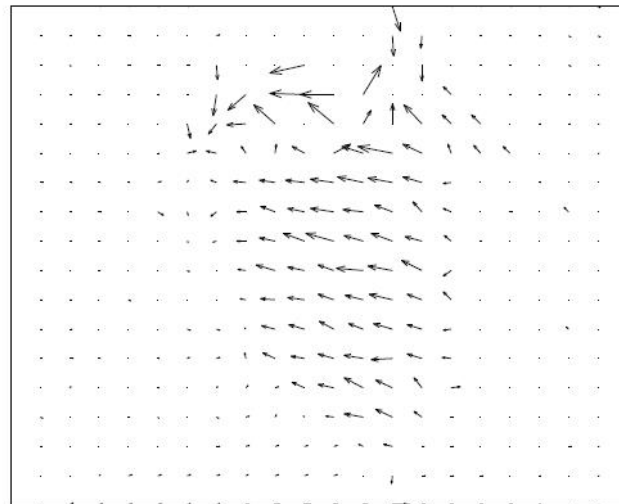
$$w(X; t_1, t_2) = X'$$

$$w(X) = X + d(X)$$

□ 流矢量（速度矢量）

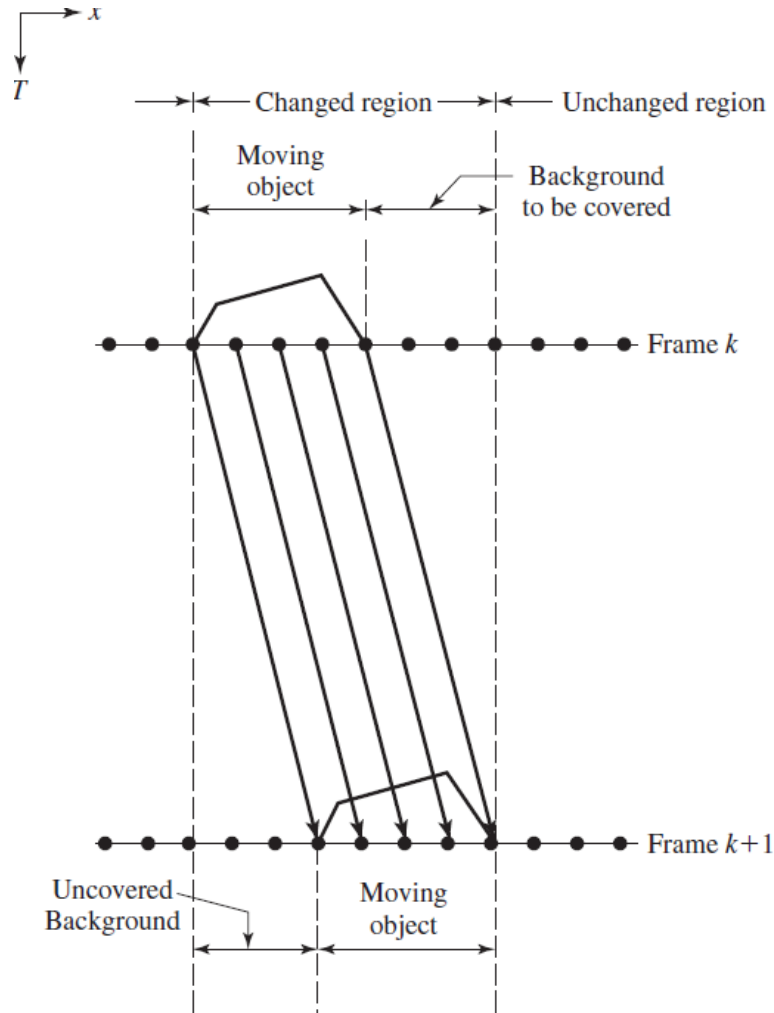
$$V = \frac{\partial d}{\partial t} = \left[\frac{\partial d_x}{\partial t}, \frac{\partial d_y}{\partial t} \right]^T$$

一个典型的二维运动场



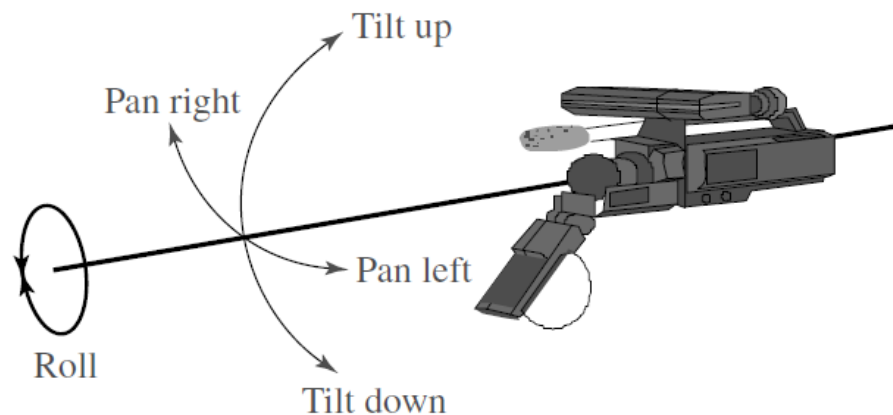
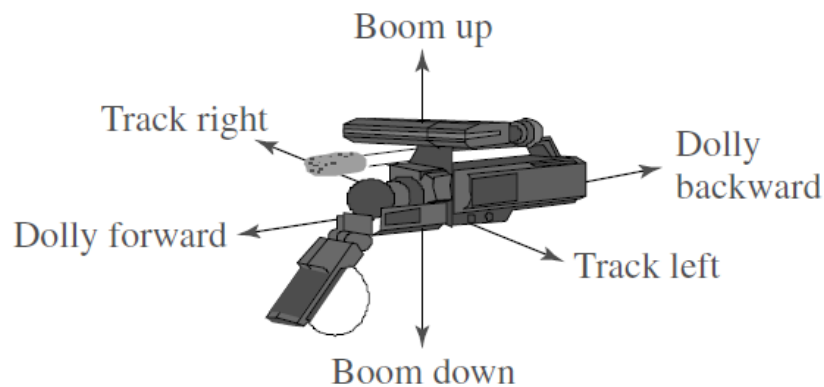
遮挡的影响

□ 在遮挡区域，运动是未定义的



典型的相机运动

□ 接下来介绍典型的相机运动对应的2D运动





相机平移：跟(track)与吊(boom)

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ 0 \end{bmatrix} \Leftrightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} FT_x / Z \\ FT_y / Z \end{bmatrix}$$

当 $\Delta Z \ll \bar{Z}$

$$\begin{bmatrix} d_x(x, y) \\ d_y(x, y) \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}, t_x = \frac{FT_x}{\bar{Z}}, t_y = \frac{FT_y}{\bar{Z}}$$



相机摇(Pan)与倾(Tilt)

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = [R_x][R_y] \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad [R_x][R_y] = \begin{bmatrix} 1 & 0 & \theta_y \\ 0 & 1 & -\theta_x \\ -\theta_y & \theta_x & 1 \end{bmatrix}$$

如果 $Y\theta_x \ll Z, X\theta_y \ll Z$, 那么 $Z' \approx Z$

$$\begin{bmatrix} d_x(x, y) \\ d_y(x, y) \end{bmatrix} = \begin{bmatrix} \theta_y F \\ -\theta_x F \end{bmatrix}$$



相机推(Zoom)和滚(Roll)

□ 推 (zoom) : 像平面与中心点距离 (焦距) 被改变

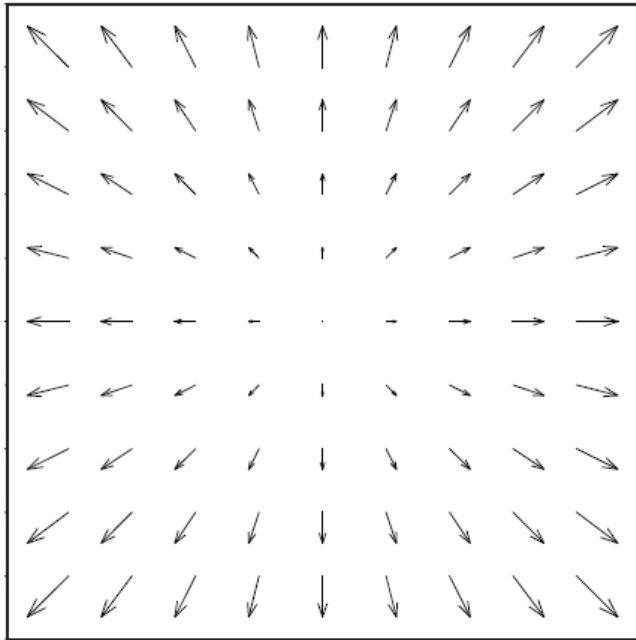
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \rho x \\ \rho y \end{bmatrix} \Rightarrow \begin{bmatrix} d_x(x, y) \\ d_y(x, y) \end{bmatrix} = \begin{bmatrix} (1 - \rho)x \\ (1 - \rho)y \end{bmatrix} \quad (\rho = F' / F)$$

□ 滚 (roll)

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta_z & -\sin \theta_z \\ \sin \theta_z & \cos \theta_z \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \approx \begin{bmatrix} 1 & -\theta_z \\ \theta_z & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

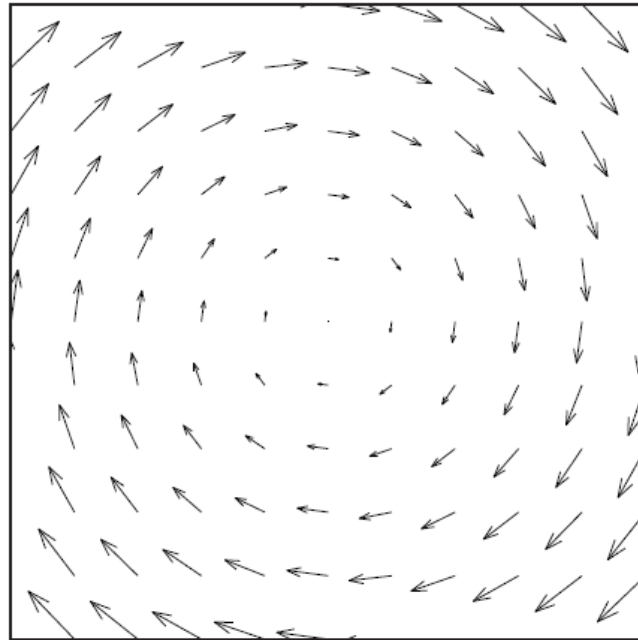
$$\begin{bmatrix} d_x(x, y) \\ d_y(x, y) \end{bmatrix} = \begin{bmatrix} -\theta_z y \\ \theta_z x \end{bmatrix}$$

相机运动的运动场



(a)

Camera zoom



(b)

Camera rotation around Z-axis (roll)

四参数模型

- 考虑一个顺序地进行平移、摇、倾、变焦和旋转的摄像机
- 几何映射：

$$\begin{aligned}\begin{bmatrix} x' \\ y' \end{bmatrix} &= \rho \begin{bmatrix} \cos \theta_z & -\sin \theta_z \\ \sin \theta_z & \cos \theta_z \end{bmatrix} \begin{bmatrix} x + \theta_y F + t_x \\ y - \theta_x F + t_y \end{bmatrix} \\ &= \begin{bmatrix} c_1 & -c_2 \\ c_2 & c_1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c_3 \\ c_4 \end{bmatrix}\end{aligned}$$

- 这个映射函数有四个参数，是仿射映射的一个特例，仿射映射一般有6个参数。



相应于三维刚性运动的二维运动模型

- 之前的相机运动模型均没有考虑相机在Z方向的平移运动
- 一般情况：

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

→
Perspective Projection

$$x' = F \frac{(r_1 x + r_2 y + r_3 F)Z + T_x F}{(r_7 x + r_8 y + r_9 F)Z + T_z F}$$
$$y' = F \frac{(r_4 x + r_5 y + r_6 F)Z + T_y F}{(r_7 x + r_8 y + r_9 F)Z + T_z F}$$



投影映射

□ 投影映射

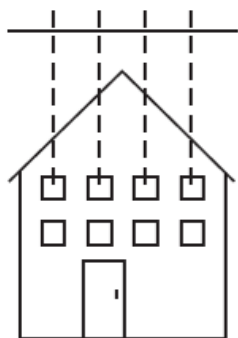
- 在Z方面没有平移运动
- 目标均有平坦表面，即 $Z = aX + bY + c$

$$x' = \frac{a_0 + a_1x + a_2y}{1 + c_1x + c_2y}, \quad y' = \frac{b_0 + b_1x + b_2y}{1 + c_1x + c_2y}$$

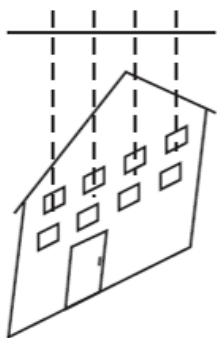
□ 实际图像可以分成多个包含平坦平面的区域

投影映射

非线性调频模型



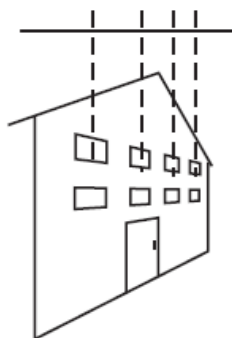
(原始的)



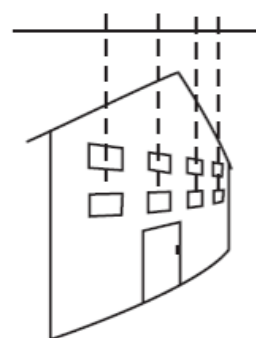
(仿射的)



(双线性的)



(投影的)



(相对投影的)



(伪透视的)



(双二次的)

线性调频模型

□ 投影映射的两个现象

- 线性调频：感知的空间频率随着与相机距离的增大而增大
- 汇聚效果：指两束平行光随着距离的增加看起来越来越远



仿射和双线性模型

□ 仿射 (6个参数):

$$\begin{bmatrix} d_x(x, y) \\ d_y(x, y) \end{bmatrix} = \begin{bmatrix} a_0 + a_1x + a_2y \\ b_0 + b_1x + b_2y \end{bmatrix}$$

- 适合将三角形映射到三角形

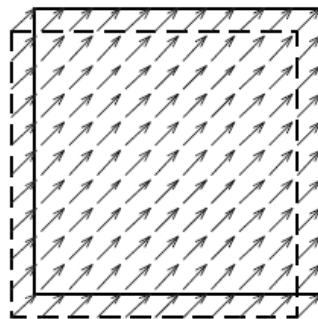
□ 双线性 (8个参数):

$$\begin{bmatrix} d_x(x, y) \\ d_y(x, y) \end{bmatrix} = \begin{bmatrix} a_0 + a_1x + a_2y + a_3xy \\ b_0 + b_1x + b_2y + b_3xy \end{bmatrix}$$

- 适合将一个四边形映射为一个曲边四边形

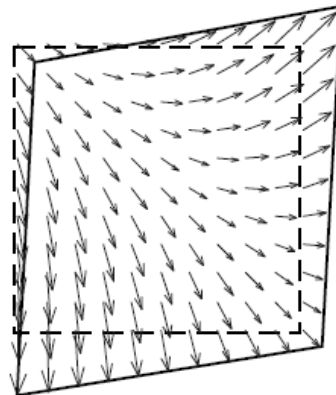
不同二维运动模型的运动场

平移



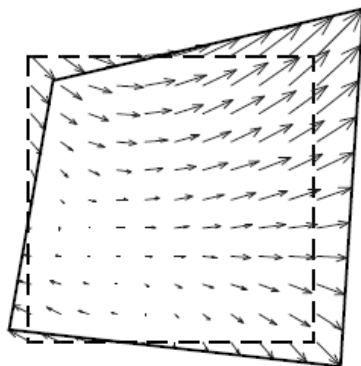
(a)

仿射



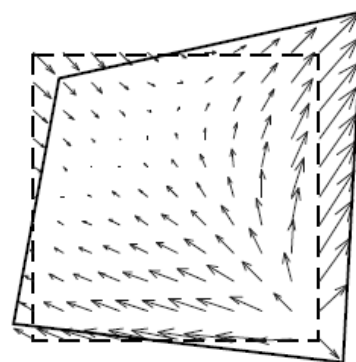
(b)

双线性



(c)

透视



(d)

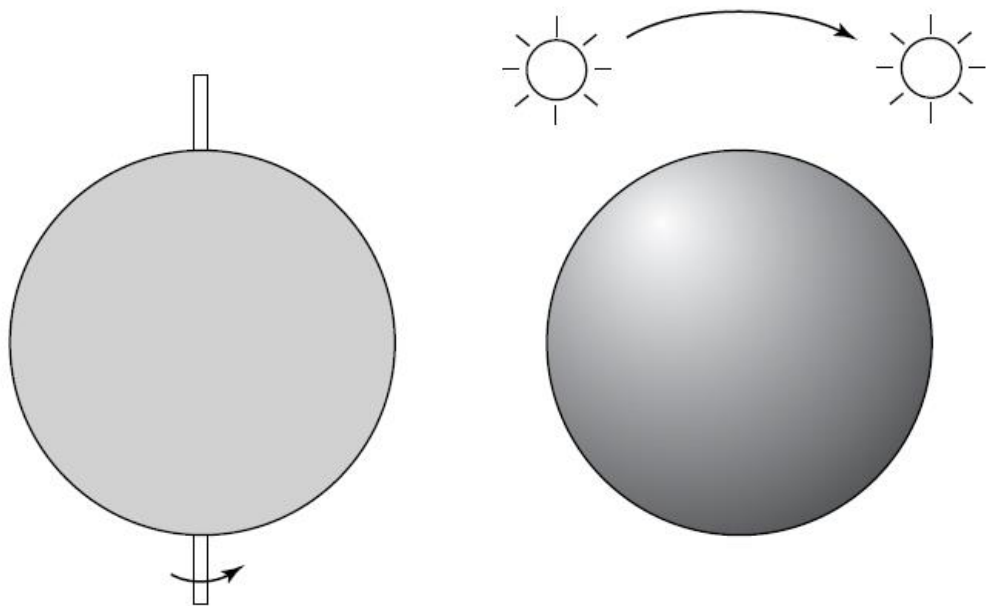


运动分析

- 二维运动模型
- **二维运动 vs. 光流**
- 运动估计中的一般方法
- 基于像素的运动估计
- 基于块的运动估计
- 多分辨率运动估计
- 相位相关法
- 可形变块匹配
- 全局运动估计
- 局部运动估计
- 运动分割

二维运动 vs. 光流

- 观察到的二维运动并不一定和实际的二维运动相同
- 仅知道图像颜色信息的情况下最好的方式就是估计光流
- 光流：基于图片模式的变化“感知”二维运动，也依赖于光照和目标表面纹理



左边：球体在恒定环境照明下转动，但是观测的图像没有变化。

右边：点绕着静止的球转动，引起球上的亮点旋转。



光流方程

- 在光照条件未知的情况下，最优的估计方法是光流估计
- 恒定亮度假设 → 光流方程

Under "constant intensity assumption":

$$\psi(x + d_x, y + d_y, t + d_t) = \psi(x, y, t)$$

But, using Taylor's expansion :

$$\psi(x + d_x, y + d_y, t + d_t) = \psi(x, y, t) + \frac{\partial \psi}{\partial x} d_x + \frac{\partial \psi}{\partial y} d_y + \frac{\partial \psi}{\partial t} d_t$$

Compare the above two, we have the optical flow equation :

$$\frac{\partial \psi}{\partial x} d_x + \frac{\partial \psi}{\partial y} d_y + \frac{\partial \psi}{\partial t} d_t = 0 \quad \text{or} \quad \frac{\partial \psi}{\partial x} v_x + \frac{\partial \psi}{\partial y} v_y + \frac{\partial \psi}{\partial t} = 0 \quad \text{or} \quad \nabla \psi^T \mathbf{v} + \frac{\partial \psi}{\partial t} = 0$$



如何使用光流方程

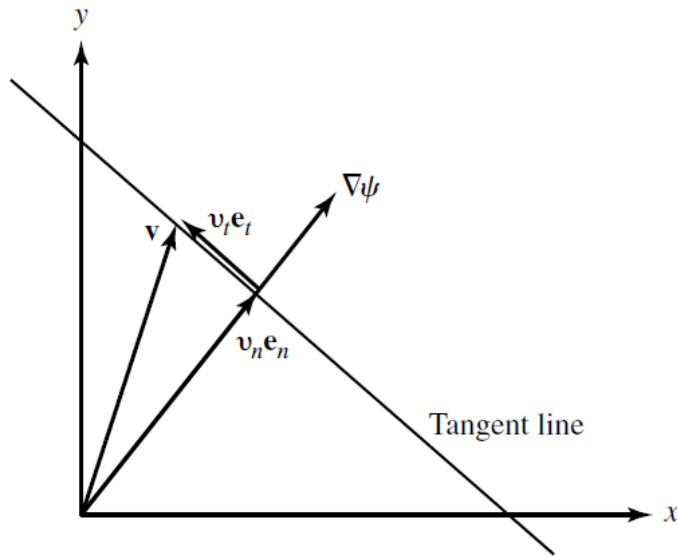
最小二乘光流估计

$$f_t \approx \frac{1}{4} [f(x, y, t+1) + f(x+1, y, t+1) + f(x, y+1, t+1) + f(x+1, y+1, t+1)] \\ - \frac{1}{4} [f(x, y, t) + f(x+1, y, t) + f(x, y+1, t) + f(x+1, y+1, t)]$$

$$f_x \approx \frac{1}{4} [f(x+1, y, t) + f(x+1, y+1, t) + f(x+1, y, t+1) + f(x+1, y+1, t+1)] \\ - \frac{1}{4} [f(x, y, t) + f(x, y+1, t) + f(x, y, t+1) + f(x, y+1, t+1)]$$

运动估计的二义性

- 光流方程仅包含梯度 v_n 方向的流向量
- 切线方向 v_t 的流向量是未定义的
- 在恒定亮度区域 $\nabla \psi = 0$ ，光流是不确定的
 - 在平坦纹理区域，运动估计是不可靠的，更可靠的是靠近边缘的区域



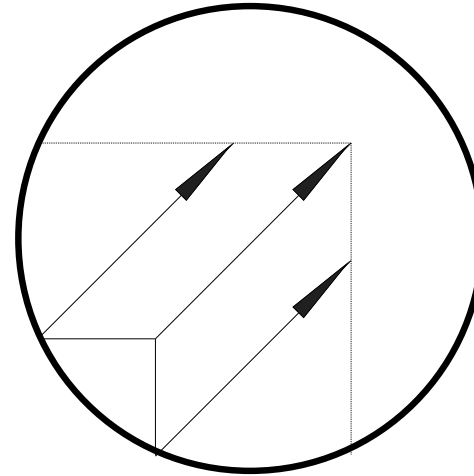
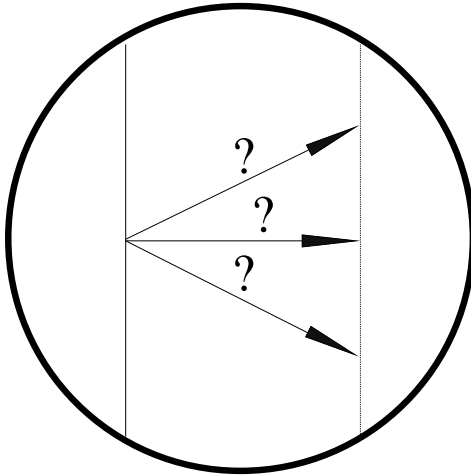
$$\nabla \psi^T \mathbf{v} + \frac{\partial \psi}{\partial t} = 0$$

$$\mathbf{v} = v_n \mathbf{e}_n + v_t \mathbf{e}_t$$

$$v_n \|\nabla \psi\| + \frac{\partial \psi}{\partial t} = 0$$

孔径问题 (Aperture problem)

<http://elvers.us/perception/aperture/>





运动分析

- ☐ 二维运动模型
- ☐ 二维运动 vs. 光流
- ☐ **运动估计中的一般方法**
- ☐ 基于像素的运动估计
- ☐ 基于块的运动估计
- ☐ 多分辨率运动估计
- ☐ 相位相关法
- ☐ 可形变块匹配
- ☐ 全局运动估计
- ☐ 局部运动估计
- ☐ 运动分割



运动估计的一般考虑

□ 两类方法：

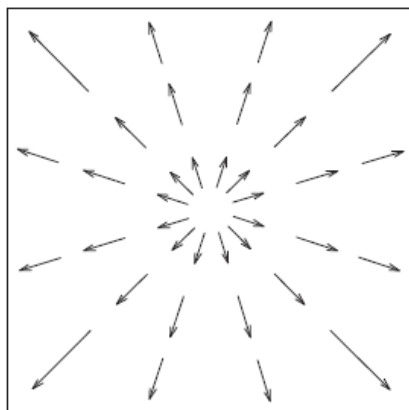
- 基于**特征**：经常用于目标跟踪，从二维重建三维目标
- 基于**亮度**（基于恒定亮度假设）：经常用于视频编码、插帧中的运动补偿预测，这也是我们关注的

□ 三个重要问题

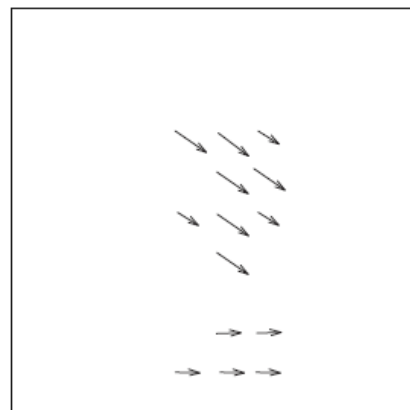
- 如何表达运动场？
- 用什么标准来估计运动参数？
- 如何搜索运动参数？

运动表达

整体：
整个运动场被一些全局
参数表达。



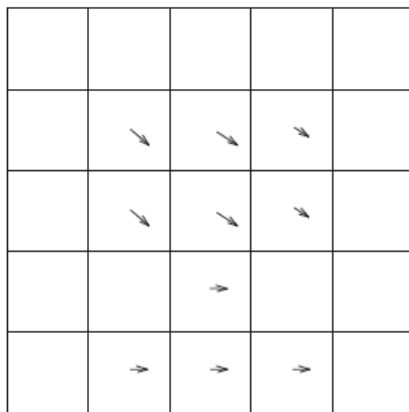
(a)



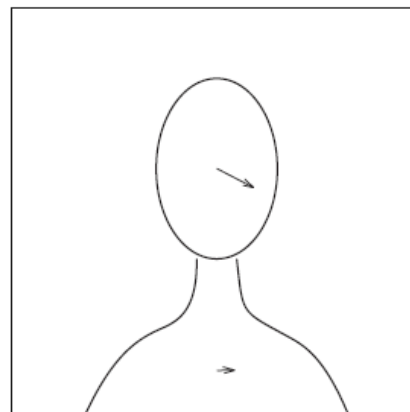
(b)

基于像素：
每一个像素有一个运动
向量，在相邻运动向量
之间有一些平滑约束。

基于块：
整个帧被分为若干个块，
每个块中的运动由一些
参数描述。



(c)

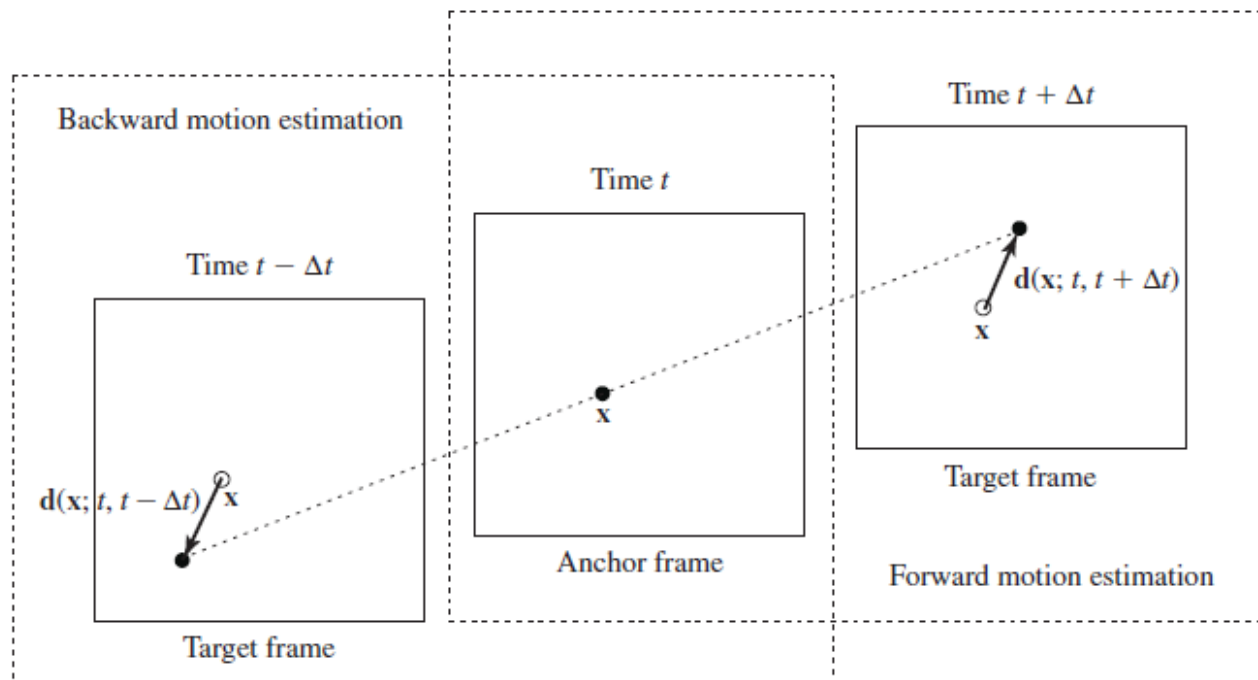


(d)

基于区域：
整帧被划分为若干区域，
每个区域对应一个具有
一致运动的目标或者子
目标，并用一些参数表
达。

其他表达：基于网格（控制网格）

符号定义



锚帧: $\psi_1(\mathbf{x})$

目标帧: $\psi_2(\mathbf{x})$

运动参数: \mathbf{a}

锚帧中一个像素点的
运动向量: $\mathbf{d}(\mathbf{x})$

运动场: $\mathbf{d}(\mathbf{x}; \mathbf{a}), \mathbf{x} \in \Lambda$

映射函数:

$\mathbf{w}(\mathbf{x}; \mathbf{a}) = \mathbf{x} + \mathbf{d}(\mathbf{x}; \mathbf{a}), \mathbf{x} \in \Lambda$



运动估计准则 (1)

□ 基于位移帧差准则 (DFD criterion)

$$E_{\text{DFD}}(\mathbf{a}) = \sum_{\mathbf{x} \in \Lambda} |\psi_2(\mathbf{x} + \mathbf{d}(\mathbf{x}; \mathbf{a})) - \psi_1(\mathbf{x})|^p \rightarrow \min$$

$$p = 1: \text{MAD}; \quad P = 2: \text{MSE}$$

$$\frac{\partial E_{\text{DFD}}}{\partial \mathbf{a}} = 2 \sum_{\mathbf{x} \in \Lambda} (\psi_2(\mathbf{w}(\mathbf{x}; \mathbf{a})) - \psi_1(\mathbf{x})) \frac{\partial \mathbf{d}(\mathbf{x})}{\partial \mathbf{a}} \nabla \psi_2(\mathbf{w}(\mathbf{x}; \mathbf{a}))$$

□ 基于光流方程准则 (OF criterion)

$$\frac{\partial \psi_1}{\partial x} d_x + \frac{\partial \psi_1}{\partial y} d_y + (\psi_2 - \psi_1) = 0 \quad \text{or} \quad \nabla \psi_1^T \mathbf{d} + (\psi_2 - \psi_1) = 0$$

$$E_{\text{OF}}(\mathbf{a}) = \sum_{\mathbf{x} \in \Lambda} \left| (\nabla \psi_2(\mathbf{x}))^T \mathbf{d}(\mathbf{x}; \mathbf{a}) + \psi_2(\mathbf{x}) - \psi_1(\mathbf{x}) \right|^p \rightarrow \min$$

$$\frac{\partial E_{\text{flow}}}{\partial \mathbf{a}} = 2 \sum_{\mathbf{x} \in \Lambda} (\nabla \psi_1(\mathbf{x})^T \mathbf{d}(\mathbf{x}; \mathbf{a}) + \psi_2(\mathbf{x}) - \psi_1(\mathbf{x})) \frac{\partial \mathbf{d}(\mathbf{x})}{\partial \mathbf{a}} \nabla \psi_1(\mathbf{x})$$



运动估计准则 (2)

- 正则化准则：利用额外的平滑项 (smoothness) 约束 (important in pixel- and block-based representation)

$$E_s(\mathbf{a}) = \sum_{\mathbf{x} \in \Lambda} \sum_{\mathbf{y} \in N_x} \|\mathbf{d}(\mathbf{x}; \mathbf{a}) - \mathbf{d}(\mathbf{y}; \mathbf{a})\|^2$$

$$w_{DFD} E_{DFD}(\mathbf{a}) + w_s E_s(\mathbf{a}) \rightarrow \min$$

- 贝叶斯准则 (Bayesian criterion)：最大化后验概率

$$P(D = \mathbf{d} | \psi_2, \psi_1) \rightarrow \max$$

$$P(\mathcal{D} = \mathbf{d} | \Psi = \psi_2; \psi_1) = \frac{P(\Psi = \psi_2 | \mathcal{D} = \mathbf{d}; \psi_1) P(\mathcal{D} = \mathbf{d}; \psi_1)}{P(\Psi = \psi_2; \psi_1)}$$



不同准则之间的联系

- OF误差准则 (OF criterion) 只有当运动较小的情况下表现良好
- 在OF误差准则下，当目标函数是MV的二次函数时，那么该函数具有封闭式解
- 当运动较大时，最好应用DFD误差准则
- 基于Bayesian准则 (Bayesian criterion) 的运动估计可以被简化为具有适当平滑约束的基于DFD的估计



优化方法

□ 穷举搜索

- 通常在DFD准则 ($p=1$) 下被采用
- 保证达到全局最优解
- 当同时搜索的参数数目很大时, 所需计算量可能是不可接受的
- 改进的快速搜索算法可以达到次优解并减少搜索时间

□ 基于梯度搜索

- 通常在DFD准则 ($p=2$) 和OF准则 ($p=2$) 下被采用
 - ✓ 梯度往往可以被解析计算得到
 - ✓ 在OF准则下, 通常可以得到封式解
- 容易得到一个接近于初始解的局部最优解, 需要通过先验知识获得一个良好的初始解

□ 多分辨率搜索策略

- 由粗到精地搜索, 比穷举搜索迅速
- 避免陷入局部最优解



运动分析

- 二维运动模型
- 二维运动 vs. 光流
- 运动估计中的一般方法
- **基于像素的运动估计**
- 基于块的运动估计
- 多分辨率运动估计
- 相位相关法
- 可形变块匹配
- 全局运动估计
- 局部运动估计
- 运动分割



基于像素的运动估计

- 运动平滑约束正则化方法
 - OF + smoothness 准则
- 多点邻域方法
 - 假设每个像素点的邻域中所有的像素都具有相同的MV
- 像素递归方法
 - 当前像素的MV是由之前已经编码的邻近像素的MV更新得到的，根据同样的更新规则，解码器可以导出同样的MV，从而MV不必编码
 - 尽管运动估计精度低，由于其简单性，被用于较早几代的视频编码器中



运动平滑约束正则化方法

□ OF + smoothness 准则

$$E(V(X)) = \sum_{X \in \Delta} \left(\frac{\partial \psi}{\partial x} v_x + \frac{\partial \psi}{\partial y} v_y + \frac{\partial \psi}{\partial t} \right)^2 + w_s (\|\nabla v_x\|^2 + \|\nabla v_y\|^2)$$

$$\nabla v_x = [v_x(x, y) - v_x(x-1, y), v_x(x, y) - v_x(x, y-1)]^T$$

$$\nabla v_y = [v_y(x, y) - v_y(x-1, y), v_y(x, y) - v_y(x, y-1)]^T$$

计算梯度的方法对算法的准确性和鲁棒性具有重要影响

用高斯预滤波加中心差分通常会得到更好的结果（sobel算子）

多点邻域方法

- 通过最小化像素的邻域像素的DFD误差，独立地估计每个像素的MV
- 假设：每个像素点的邻域中所有的像素都具有相同的MV
- 目标函数：

$$E_{\text{DFD}}(\mathbf{d}_n) = \sum_{\mathbf{x} \in B(\mathbf{x}_n)} w(\mathbf{x}) |\psi_2(\mathbf{x} + \mathbf{d}_n) - \psi_1(\mathbf{x})|^2 \rightarrow \min$$

- 优化方法：
 - 穷举搜索 (假设每次只需要求解一个MV)
 - ✓ 需要选择合适的搜索范围和搜索步长
 - 基于梯度搜索



运动分析

- ☐ 二维运动模型
- ☐ 二维运动 vs. 光流
- ☐ 运动估计中的一般方法
- ☐ 基于像素的运动估计
- ☐ **基于块的运动估计**
- ☐ 多分辨率运动估计
- ☐ 相位相关法
- ☐ 可形变块匹配
- ☐ 全局运动估计
- ☐ 局部运动估计
- ☐ 运动分割



块匹配算法

- 假设一个块内所有像素都具有一致的运动，即可独立地估计每个块的运动参数
- 块匹配算法 (BMA): 仅**平移运动**, 对每个块估计一个MV (1 MV, 2 parameter)
 - 穷举BMA (EBMA)
 - 快速算法



块匹配算法 (BMA)

□ 概述：

- 假设块中所有像素仅有同一个平移运动，用一个MV即可表示
- 通过最小化块中的DFD误差，估计MV

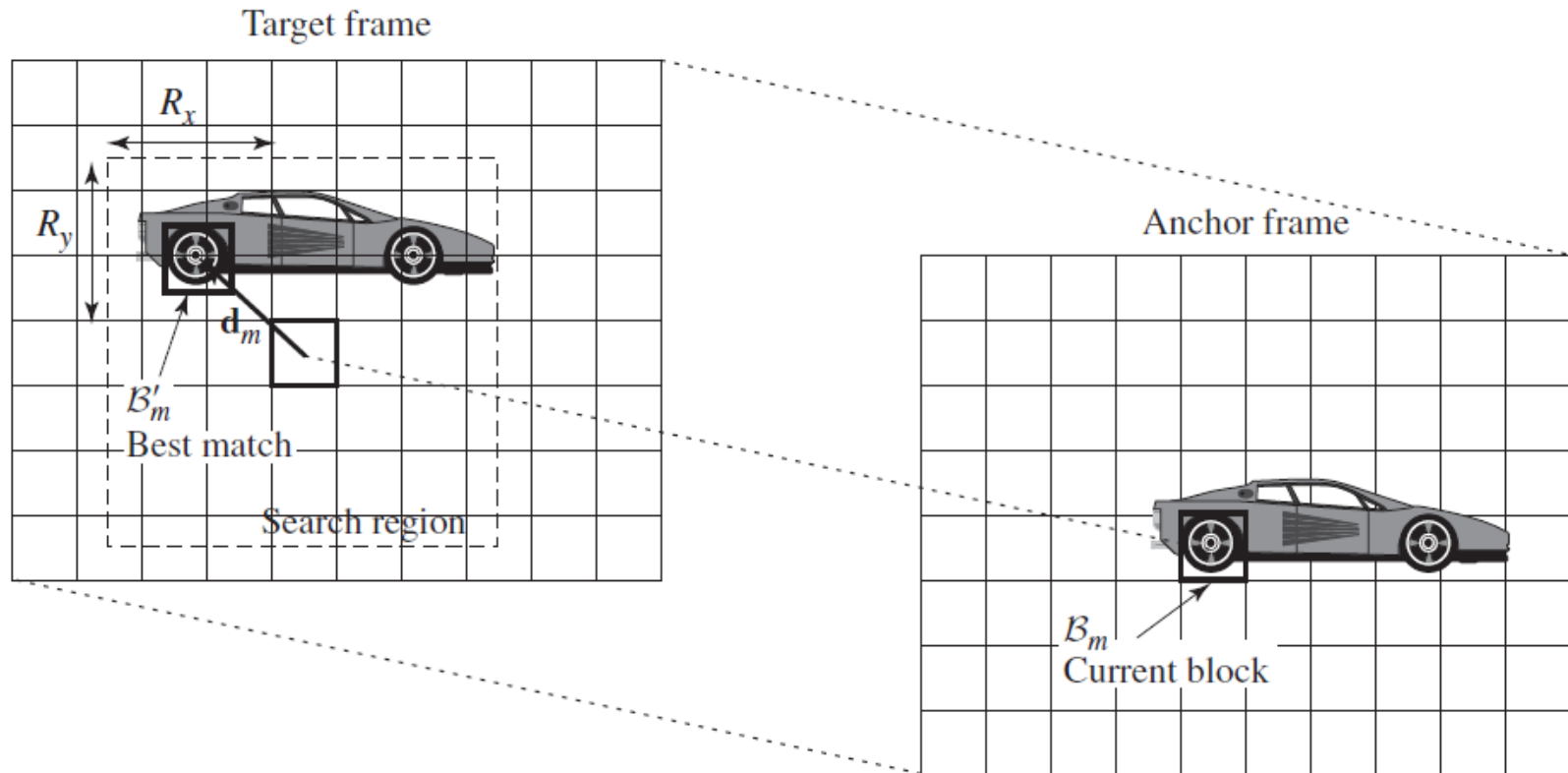
□ 目标函数：

$$E_{\text{DFD}}(\mathbf{d}_m) = \sum_{\mathbf{x} \in B_m} |\psi_2(\mathbf{x} + \mathbf{d}_m) - \psi_1(\mathbf{x})|^p \rightarrow \min$$

□ 优化方法：

- 穷举搜索
 - ✓ 每次只需要求解一个MV
 - ✓ 可使用MAD准则，即 $p=1$
- 快速搜索算法
- 整数精度搜索 vs. 分数精度搜索

穷举BMA (EBMA)





整数像素精度EBMA复杂度

□ 假设

- 图像尺寸: $M \times M$
- 块尺寸: $N \times N$
- 搜索范围: $(-R, R)$ in each dimension
- 搜索步长: 1 pixel (assuming integer MV)

□ 操作数 (Operation counts):

(1 operation=1 “-”, 1 “abs”, 1 “+”)

- 每个候选位置的像素灰度比较数: N^2
- 每个参考块需要遍历的候选位置: $(2R + 1)^2$
- 整一帧: $(M/N)^2 (2R + 1)^2 N^2 = M^2 (2R + 1)^2$
✓ 独立于块尺寸!

□ 例子: $M=512$, $N=16$, $R=16$, 30 fps

- 总操作数 = $2.85 \times 10^8 / \text{frame} \times 30 \text{ frame/s} = 8.55 \times 10^9 / \text{s}$

□ 适用于超大规模集成电路 (VLSI) 进行实现

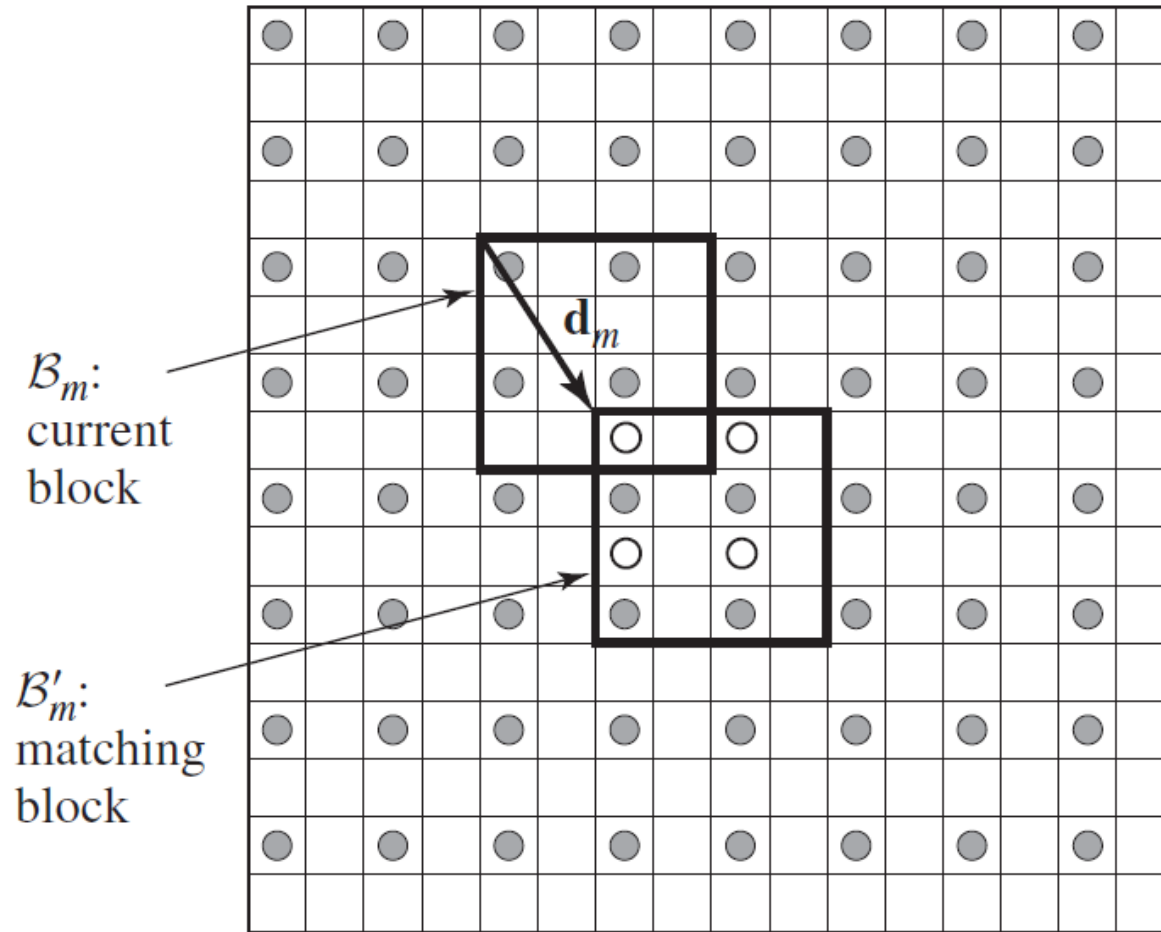
- 软件实现困难



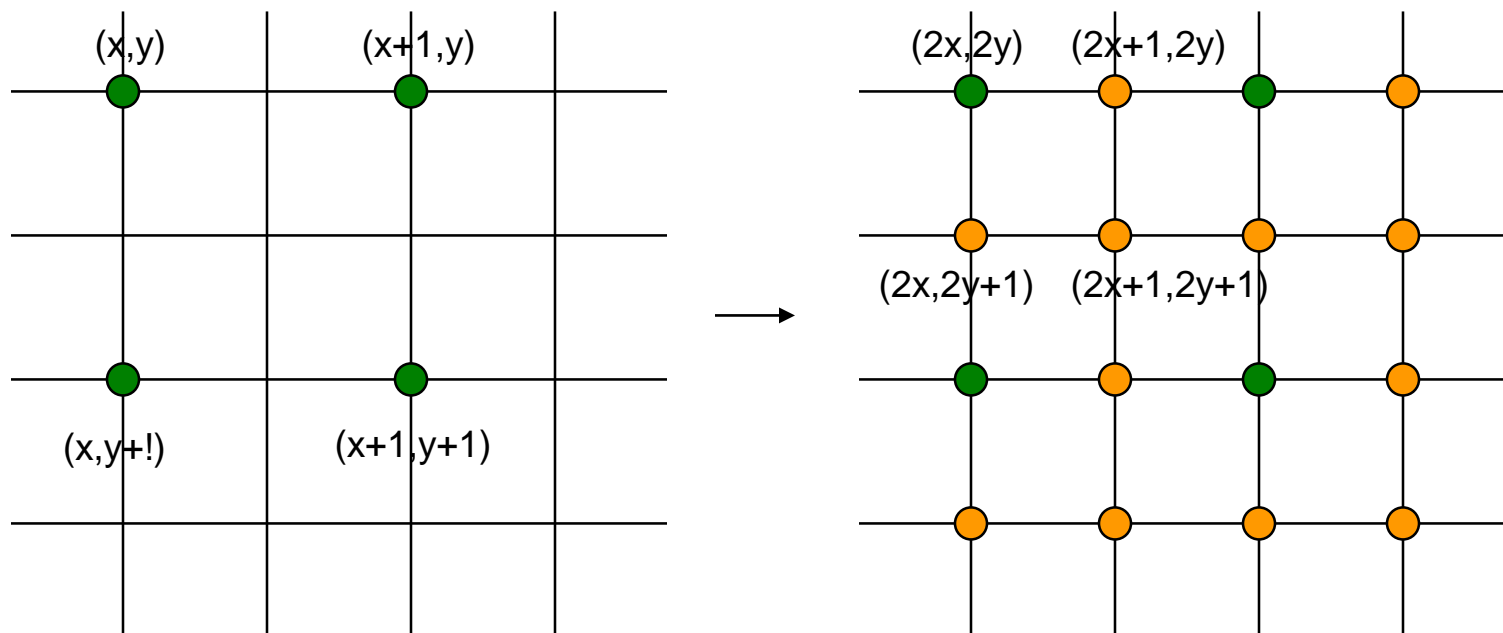
分数像素精度EBMA

- MV估计中，搜索步长并不一定是一个整数，在实际情况下，分数步长可能更合适
- 半像素精度EBMA: step-size=1/2 pixel in both dimension
- 困难：
 - 目标帧仅有整数像素点
- 解决方案：
 - 在搜索之前目标帧先进行2倍内插
 - 通常采用双线性插值
- 计算复杂度：
 - 4倍于整数像素精度，并加上额外的插值开销
- 快速算法：
 - 首先以整数精度进行搜索，然后在小范围内以半像素精度进行细化

半像素精度EBMA



双线性插值



$$O[2x,2y]=I[x,y]$$

$$O[2x+1,2y]=(I[x,y]+I[x+1,y])/2$$

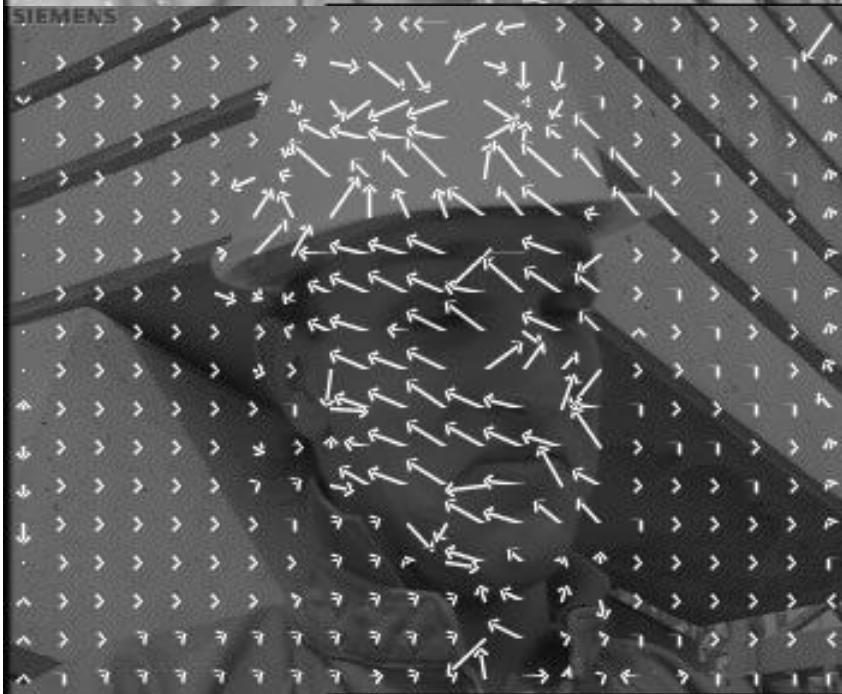
$$O[2x,2y+1]=(I[x,y]+I[x+1,y])/2$$

$$O[2x+1,2y+1]=(I[x,y]+I[x+1,y]+I[x,y+1]+I[x+1,y+1])/4$$

target frame



Motion field



anchor frame



Predicted anchor frame (29.86dB)



Example: 半像素精度EBMA



BMA快速算法

□ 如何减少EBMA计算量？

■ 降低搜索候选块的数量：

- ✓ 只搜索那些可能产生小误差的块
- ✓ 根据之前的搜索结果，预测可能剩下的候选块

■ 简化误差度量准则 (DFD)

□ 经典的快速算法

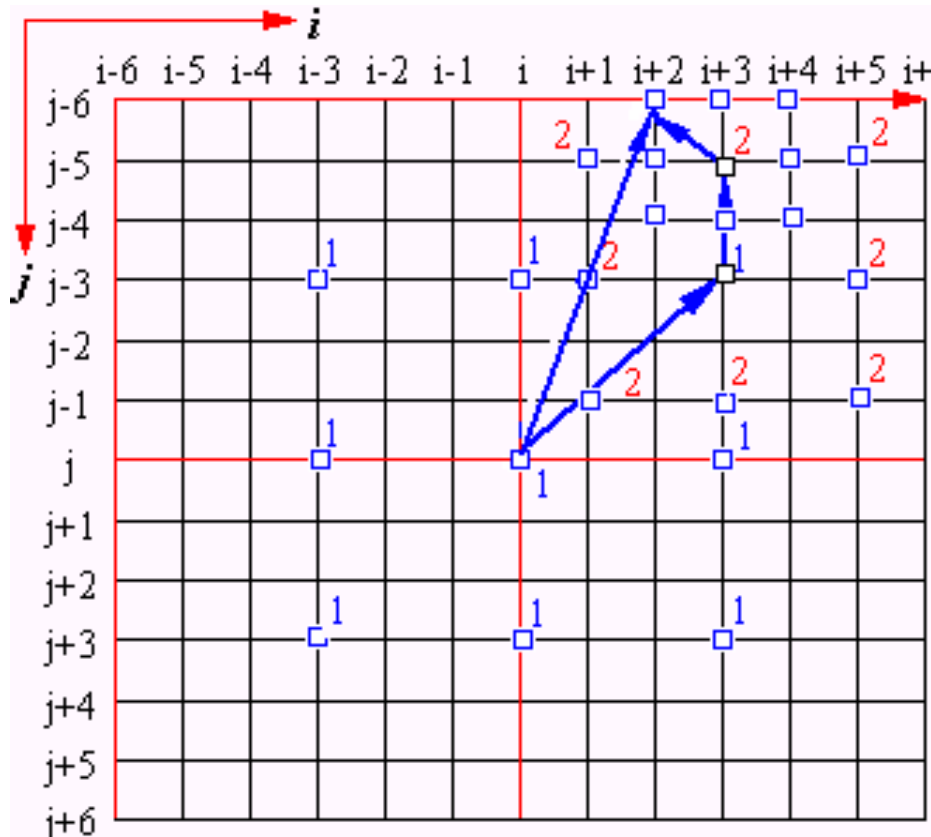
■ 三步搜索法 (Three-step)

■ 二维对数搜索法 (2D-log)

□ 还有许多新的快速算法

■ 有些适合软件实现，有些适合VLSI实现

三步搜索法



R_0 : initial search step

Search step L

$$L = \lfloor \log_2 R_0 + 1 \rfloor$$

Total number: $8L+1$

For example

$R=32$

EBMA: $4225 = (2R+1)^2$

3Step: $41 = 8*5+1$

EBMA存在的问题(I)

□ 块效应 (块边界的不连续性)

- 基于块的平移运动模型不准确
- 实际的运动情况比平移更复杂
 - ✓ 解决方案: 可形变的BMA (deformable BMA)
- 在一个块中可能有多个具有不同运动的对象
 - ✓ 解决方案:
 - 基于区域的运动估计
 - 基于网格模型的运动估计
- 光照影响
 - ✓ 进行光照补偿以满足“恒定光强假设”



EBMA存在的问题 (II)

□ 运动场混乱

- 原因：逐块**独立**地估计MV
- 解决方案：
 - ✓ 加入显式的**平滑约束项**
 - ✓ 多分辨率方法
 - ✓ 基于网格模型的运动估计

□ 平坦区的MV预测出错

- 当空间上梯度接近于零时，运动难以确定
- 应该使用非规则的理想的分块
- 解决方案：基于区域的运动估计

□ 需要巨大的计算量

- 解决方案：
 - ✓ 快速算法：多分辨率方法



运动分析

- ☐ 二维运动模型
- ☐ 二维运动 vs. 光流
- ☐ 运动估计中的一般方法
- ☐ 基于像素的运动估计
- ☐ 基于块的运动估计
- ☐ **多分辨率运动估计**
- ☐ 相位相关法
- ☐ 可形变块匹配
- ☐ 全局运动估计
- ☐ 局部运动估计
- ☐ 运动分割



多分辨率运动估计

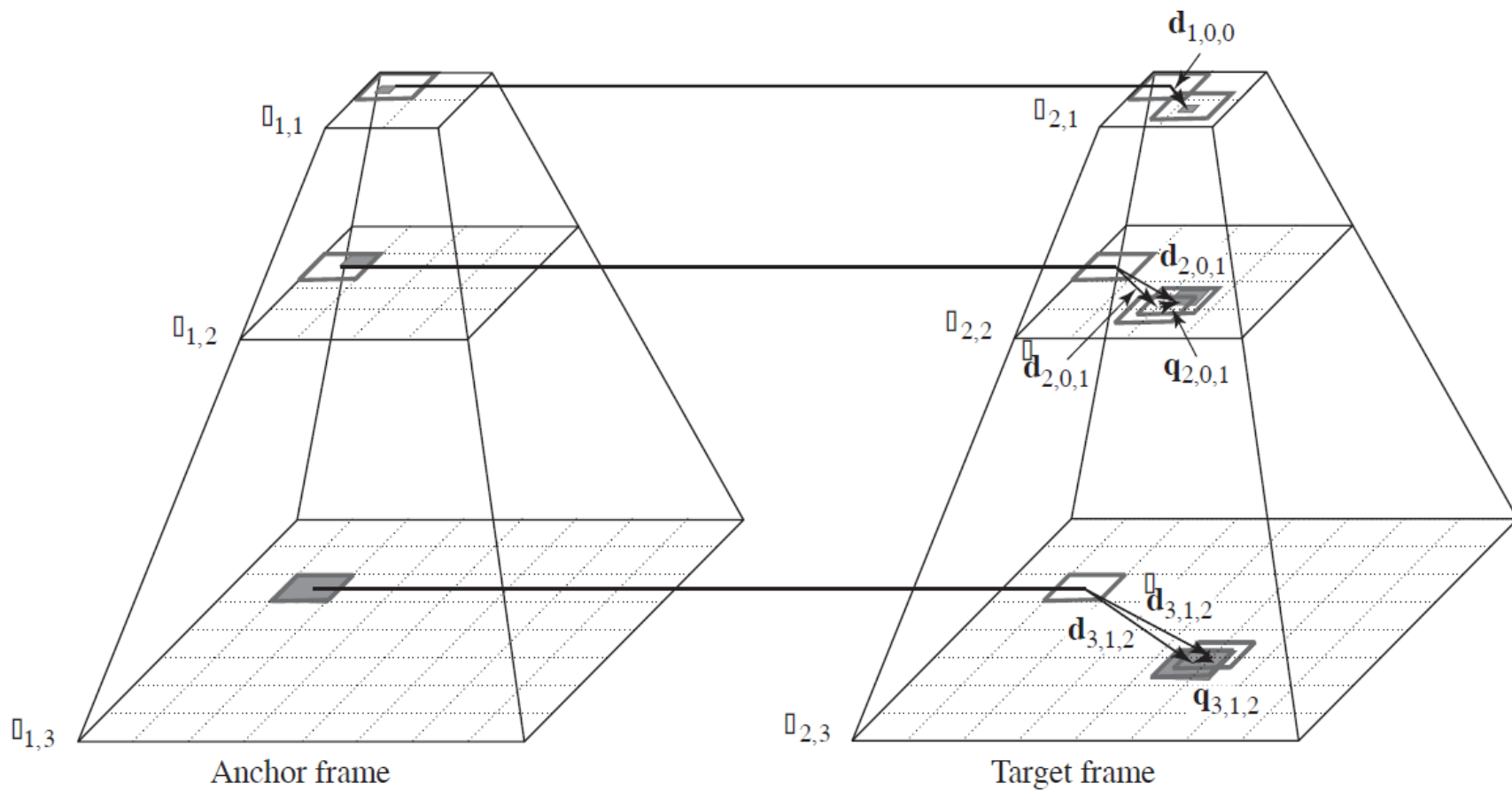
□ BMA存在缺陷

- 除非使用穷举搜索，否则可能难以达到全局最优解
- 穷举搜索需要非常大的计算量
- 基于块的平移运动模型并不总是合适的

□ 多分辨率估计方法

- 解决上述前两个问题
- 首先在低通滤波、下采样的图像对上，进行低分辨率下的运动估计
 - ✓ 通常能得到一个接近于真实运动场的解
- 然后在较小的搜索范围内以更高的分辨率逐步改善初始解
 - ✓ 降低计算量
- 可以应用于不同的运动场景下，后续内容中我们只集中介绍其在BMA中的应用

多分辨率运动估计





分层块匹配算法 (HBMA)

Number of levels: L

t th level image: $\Psi_{t,l}(X), X \in \Lambda_l, t = 1, 2$

Interpolation operator: $\tilde{d}_l(X) = \mathcal{U}(d_{l-1}(X))$

Error function: $\sum_{X \in \Lambda_l} |\Psi_{2,l}(X + \tilde{d}_l(X) + q_l(X)) - \Psi_{1,l}(X)|^p$

Update motion vector: $d_l(X) = \tilde{d}_l(X) + q_l(X)$

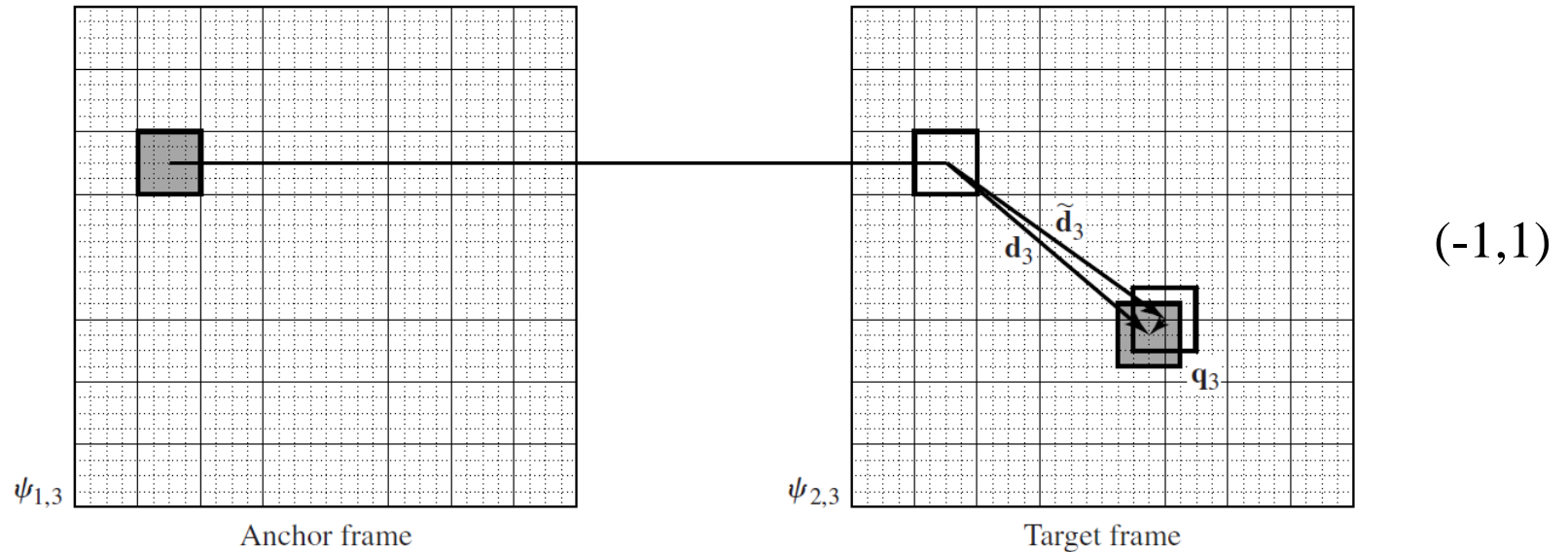
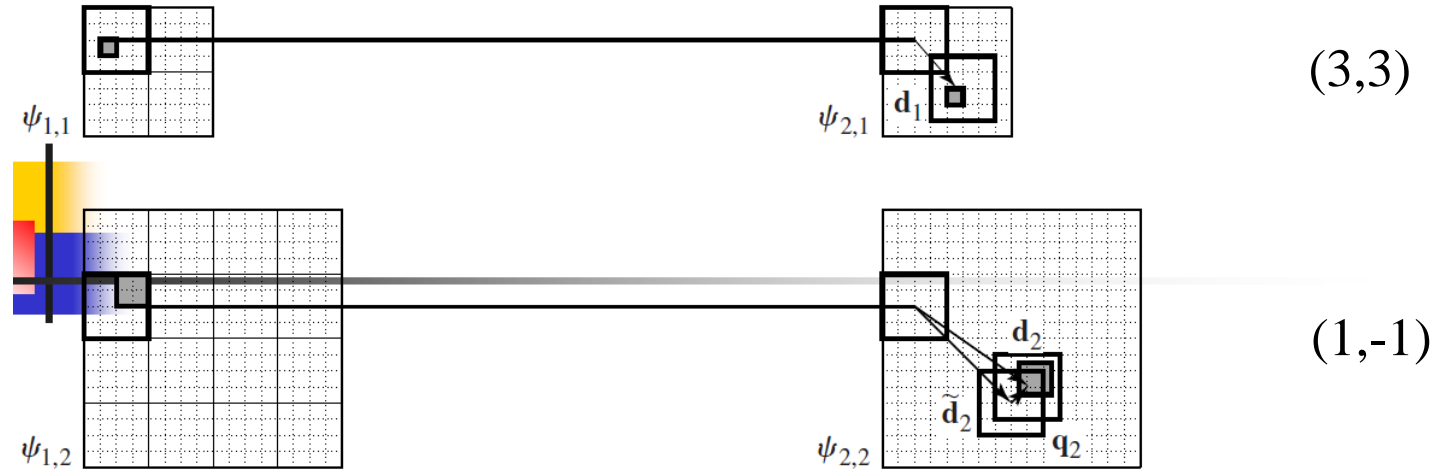
MV at l th level prediction:

$$\tilde{d}_{l,m,n}(X) = \mathcal{U}(d_{l-1, \lfloor m/2 \rfloor, \lfloor n/2 \rfloor}(X)) = 2d_{l-1, \lfloor m/2 \rfloor, \lfloor n/2 \rfloor}(X)$$

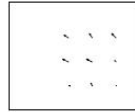
Total motion:

$$d_l(X) = q_L(X) + \mathcal{U}(q_{L-1}(X) + \mathcal{U}(q_{L-2}(X) \cdots + \mathcal{U}(q_1(X) + d_0(X)) \cdots))$$

分层块匹配算法 (HBMA)



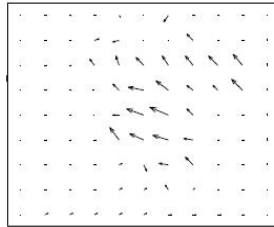
分层块匹配算法 (HBMA)



(a)



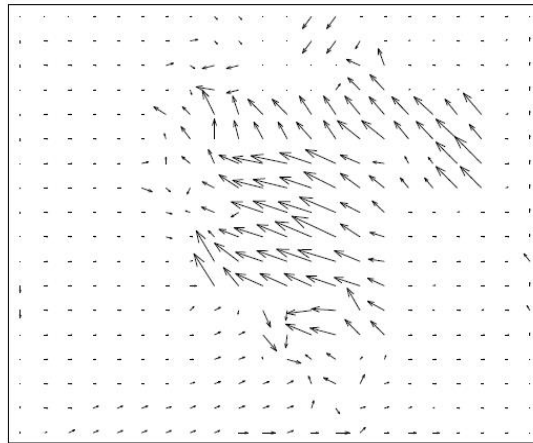
(b)



(c)



(d)



(e)



(f)

Example: Three-level HBMA

Predicted anchor frame (29.32dB)



HBMA复杂度

□ 假设

- 图像尺寸: $M \times M$
- 块尺寸: $N \times N$ at every level; Levels: L
- 搜索范围:
 - ✓ 1st level: $R/2^{(L-1)}$ (Equivalent to R in L -th level)
 - ✓ Other levels: $R/2^{(L-l)}$ (can be smaller)

□ EBMA

- 图像尺寸 = $M \times M$, 块尺寸 = $N \times N$, 搜索范围 = $(-R, R)$
- 操作数: $M^2(2R+1)^2$

□ HBMA l -th level 操作数 (图像尺寸: $M/2^{L-l}$)

$$\left(M/2^{L-l}\right)^2 \left(2R/2^{L-l} + 1\right)^2$$

□ HBMA总操作数

$$\sum_{l=1}^L \left(M/2^{L-l}\right)^2 \left(2R/2^{L-l} + 1\right)^2 \approx \frac{1}{3} 4^{-(L-2)} 4M^2 R^2$$

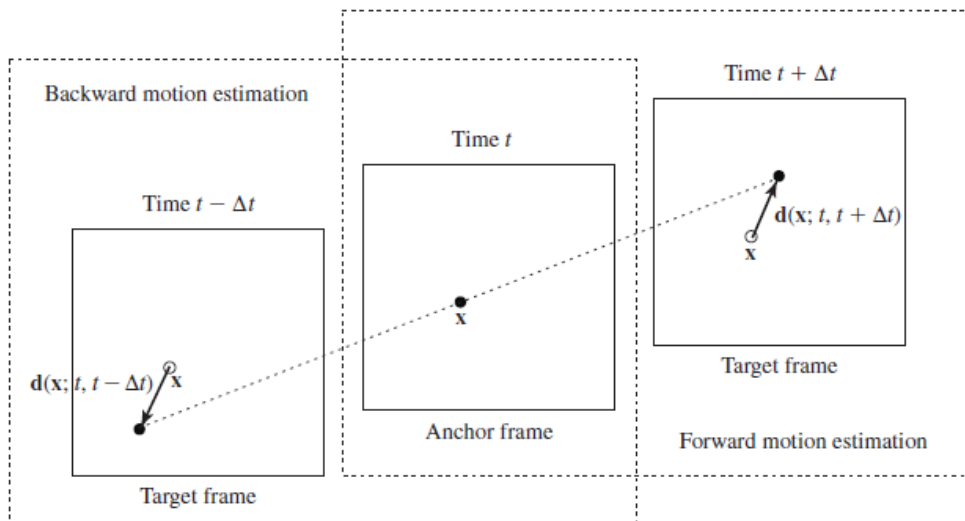
□ EBMA / HBMA: $3 \cdot 4^{(L-2)} = 3(L=2); 12(L=3)$



运动分析

- ☐ 二维运动模型
- ☐ 二维运动 vs. 光流
- ☐ 运动估计中的一般方法
- ☐ 基于像素的运动估计
- ☐ 基于块的运动估计
- ☐ 多分辨率运动估计
- ☐ **相位相关法**
- ☐ 可形变块匹配
- ☐ 全局运动估计
- ☐ 局部运动估计
- ☐ 运动分割

相位相关法



识别相位相关函数的峰值 (PCF)

$$\psi_1(X) = \psi_2(X + d)$$

$$\bar{\psi}_1(f) = \bar{\psi}_2(f) \cdot e^{j2\pi d^T f}$$

$$\tilde{\psi}(f) = \frac{\bar{\psi}_1(f) \cdot \bar{\psi}_2^*(f)}{|\bar{\psi}_2(f) \cdot \bar{\psi}_2^*(f)|} = e^{j2\pi d^T f}$$

$$PCF(X) = F^{-1}\{\tilde{\psi}(f)\} = \delta(X + d)$$

□ Note

- 减轻边界采样效应：空间域加权窗函数
- 广泛应用于图像配准
- 优点：对光照变化不敏感

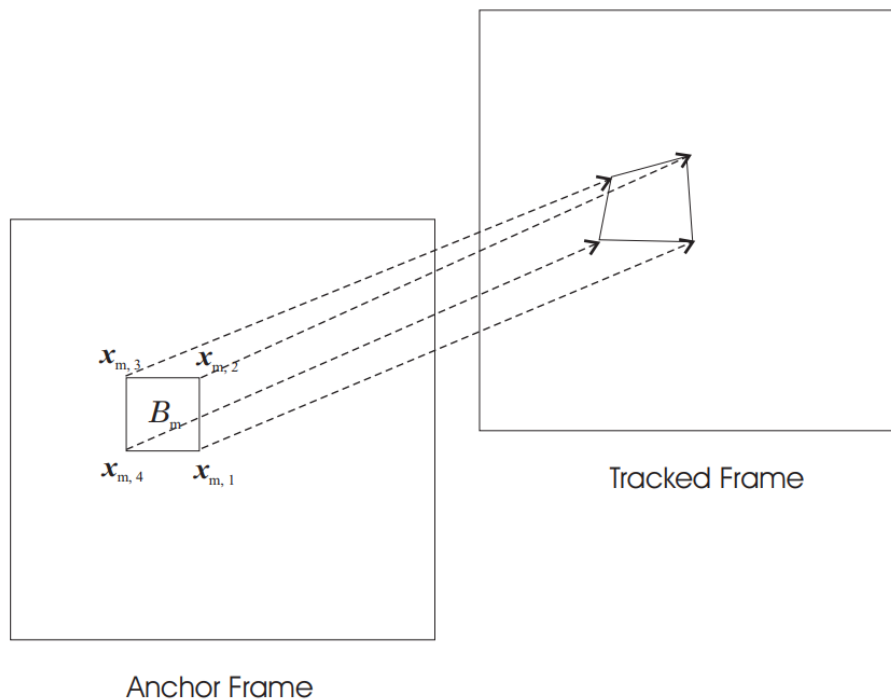


运动分析

- 二维运动模型
- 二维运动 vs. 光流
- 运动估计中的一般方法
- 基于像素的运动估计
- 基于块的运动估计
- 多分辨率运动估计
- 相位相关法
- 可形变块匹配
- 全局运动估计
- 局部运动估计
- 运动分割

可形变块匹配

- 之前的图像块匹配算法主要考虑平移运动，无法刻画旋转、缩放、仿射等更高阶的运动
- 可形变块匹配算法
 - 面向更高阶运动的图像块匹配算法

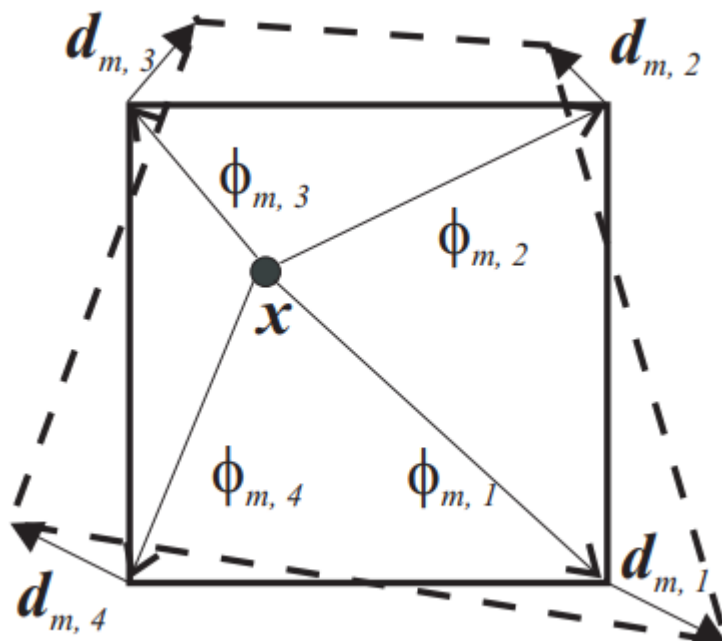


可形变块匹配

□ 基于顶点的运动信息表征

- 基于参数的运动模型不适合与现有编码框架

$$\mathbf{d}_m(\mathbf{x}) = \sum_{k=1}^K \phi_{m,k}(\mathbf{x}) \mathbf{d}_{m,k}, \quad \mathbf{x} \in \mathcal{B}_m.$$



可形变块匹配

□ 运动估计方法

$$E(\mathbf{a}) = \sum_{\mathbf{x} \in \mathcal{B}} |\psi_2(\mathbf{w}(\mathbf{x}; \mathbf{a})) - \psi_1(\mathbf{x})|^p$$
$$\mathbf{w}(\mathbf{x}; \mathbf{a}) = \mathbf{x} + \sum_{k \in \mathcal{K}} \phi_k(\mathbf{x}) \mathbf{d}_k$$

□ 基于梯度的方法

$$\frac{\partial E}{\partial \mathbf{a}}(\mathbf{a}) = \left[\frac{\partial E}{\partial \mathbf{a}_x}(\mathbf{a}), \frac{\partial E}{\partial \mathbf{a}_y}(\mathbf{a}) \right]^T,$$

$$\frac{\partial E}{\partial \mathbf{a}_x}(\mathbf{a}) = 2 \sum_{x \in \mathcal{B}} e(\mathbf{x}; \mathbf{a}) \frac{\partial \psi_2(\mathbf{w}(\mathbf{x}; \mathbf{a}))}{\partial x} \phi(\mathbf{x}),$$
$$\frac{\partial E}{\partial \mathbf{a}_y}(\mathbf{a}) = 2 \sum_{x \in \mathcal{B}} e(\mathbf{x}; \mathbf{a}) \frac{\partial \psi_2(\mathbf{w}(\mathbf{x}; \mathbf{a}))}{\partial y} \phi(\mathbf{x}).$$



运动分析

- ☐ 二维运动模型
- ☐ 二维运动 vs. 光流
- ☐ 运动估计中的一般方法
- ☐ 基于像素的运动估计
- ☐ 基于块的运动估计
- ☐ 多分辨率运动估计
- ☐ 相位相关法
- ☐ 可形变块匹配
- ☐ **全局运动估计**
- ☐ 局部运动估计
- ☐ 运动分割



全局运动估计

- 全局运动：
 - 摄像机在固定的场景上移动
 - ✓ 大多数投影摄像机的运动都可以用仿射映射来表达
 - 整个场景都在移动: 鲜有发生
 - 通常，场景可以分解为几个主要区域，每个区域的移动方式不同 (基于场景的运动估计)
- 如果存在一个全局运动，或者区域运动具有运动一致性，我们可以估计出运动参数
 - 直接估计
 - 间接估计
- 当绝大多数像素 (非全部) 具有运动一致性，我们可以迭代地估计出运动参数以及得到所对应的像素点
 - 稳健估计
 - Inlier and outlier

直接估计

- 使用运动参数，将DFD误差表示为如下形式，并通过最小化DFD误差来估计这些参数

$$E_{\text{DFD}} = \sum_{n \in \mathcal{N}} w_n |\psi_2(\mathbf{x}_n + \mathbf{d}(\mathbf{x}_n; \mathbf{a})) - \psi_1(\mathbf{x}_n)|^p$$

w_n 是 \mathbf{x}_n 的加权系数，取决于运动估计在 \mathbf{x}_n 的精度

Ex: 仿射运动:

$$\begin{bmatrix} d_x(\mathbf{x}_n; \mathbf{a}) \\ d_y(\mathbf{x}_n; \mathbf{a}) \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x_n + a_2 y_n \\ b_0 + b_1 x_n + b_2 y_n \end{bmatrix}, \quad \mathbf{a} = [a_0, a_1, a_2, b_0, b_1, b_2]^T$$

- 穷举搜索或梯度下降法都可以用来找到一个最小的 E_{DFD}

间接估计

- 首先使用基于像素或基于块的方法找到深度运动场(e.g. EBMA)
- 然后通过最小二乘拟合，利用运动模型对得到的运动场进行参数化

$$E_{fit} = \sum w_n (\mathbf{d}(\mathbf{x}_n; \mathbf{a}) - \mathbf{d}_n)^2$$

Affine motion :

$$\mathbf{d}(\mathbf{x}_n; \mathbf{a}) = [\mathbf{A}_n] \mathbf{a},$$

$$[\mathbf{A}_n] = \begin{bmatrix} 1 & x_n & y_n & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_n & y_n \end{bmatrix}$$

$$\frac{\partial E_{fit}}{\partial \mathbf{a}} = \sum w_n [\mathbf{A}_n]^T ([\mathbf{A}_n] \mathbf{a} - \mathbf{d}_n) = 0$$

$$\mathbf{a} = \left(\sum w_n [\mathbf{A}_n]^T [\mathbf{A}_n] \right)^{-1} \left(\sum w_n [\mathbf{A}_n]^T \mathbf{d}_n \right)$$

w_n 是 \mathbf{x}_n 的加权系数，
取决于运动估计在 \mathbf{x}_n 的精度。



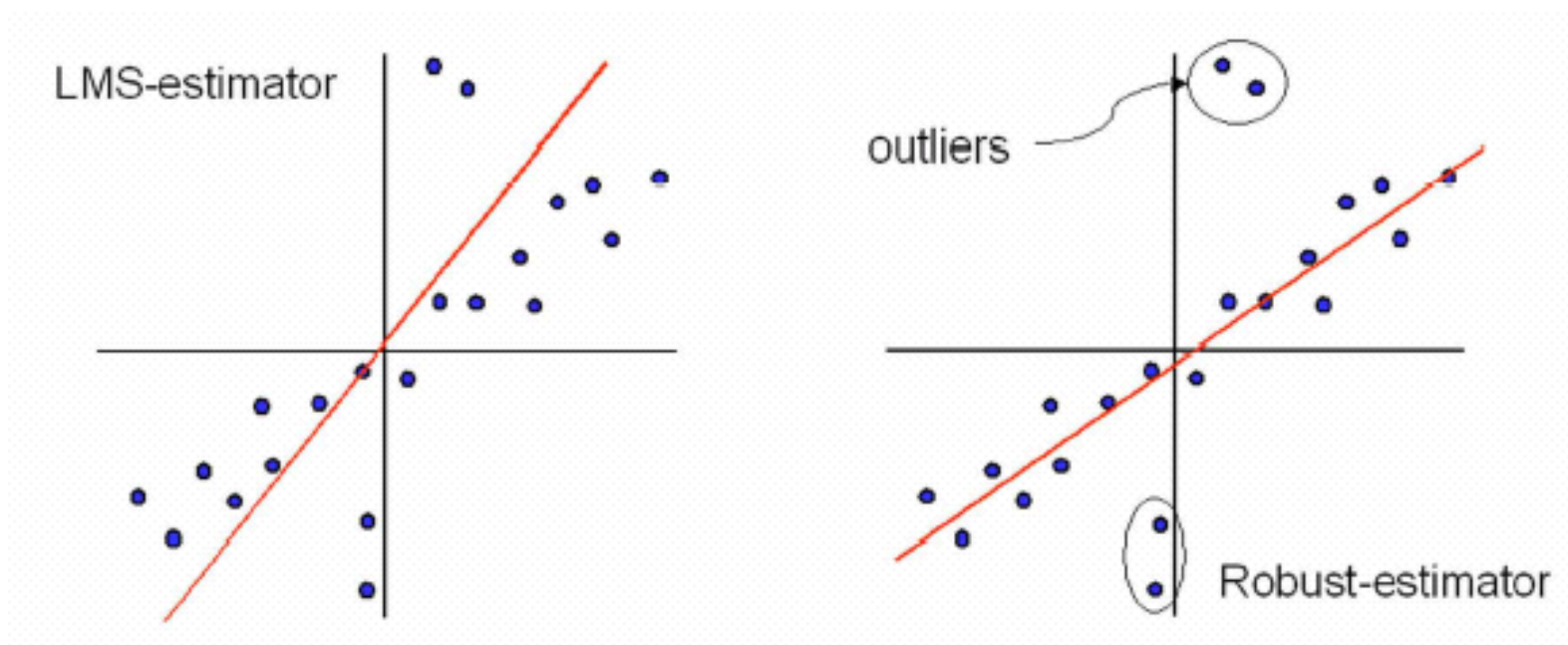
稳健估计

□ 实质: 迭代删除离群点(outlier)像素

1. 将区域设置为帧中的所有像素
2. 将直接或间接方法应用于区域内的所有像素
3. 评估区域内所有像素的误差(E_{DFD} or E_{fit})
4. 删除有较大错误的离群像素
5. 对区域中的其余像素重复步骤2-4

细节: 硬阈值与软阈值

稳健估计



使用LMS(最小均方)和稳健估计器拟合数据点的直线



运动分析

- 二维运动模型
- 二维运动 vs. 光流
- 运动估计中的一般方法
- 基于像素的运动估计
- 基于块的运动估计
- 多分辨率运动估计
- 相位相关法
- 可形变块匹配
- 全局运动估计
- **局部运动估计**
- 运动分割



局部运动估计：基于区域的运动估计

□ 基本假设

- 场景由多个对象组成，每个对象(或子对象)对应的区域的运动具有一致性。
- 从物理意义上考虑，基于区域的运动估计比基于块、基于网格或全局运动模型更正确

□ 方法:

- 区域优先: 基于纹理/边缘将帧分割成多个区域，然后使用全局运动估计方法估计每个区域的运动
- 运动优先: 估计整个图像的运动场，然后对运动场进行分割，这样每个区域的运动就可以用一组参数精确地描述
- 对区域分割和每一个区域的运动进行联合估计: 迭代交替的进行区域分割与运动估计



运动分析

- ☐ 二维运动模型
- ☐ 二维运动 vs. 光流
- ☐ 运动估计中的一般方法
- ☐ 基于像素的运动估计
- ☐ 基于块的运动估计
- ☐ 多分辨率运动估计
- ☐ 相位相关法
- ☐ 可形变块匹配
- ☐ 全局运动估计
- ☐ 局部运动估计
- ☐ **运动分割**



运动分割

- 为什么要对物体的运动进行分割
 - 帮助改善多运动光流估计
 - 帮助改善三维运动和结构估计
 - 目标识别
 - 目标跟踪
 - 基于对象的视频编码
 - 基于对象的编辑(合成变形)

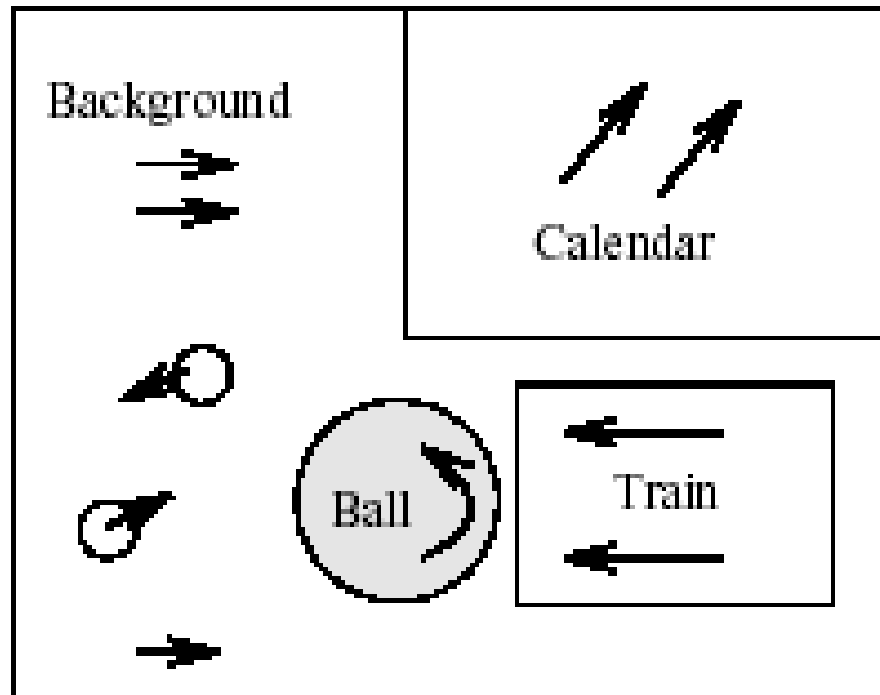


图像分割 vs 运动分割

- 分割是基于一个特征(向量)的操作
 - e.g. 图像分割通常基于灰度、颜色或纹理
- 将图像分割方法直接应用于运动分割(比如使用速度向量作为特征)可能并不有效, 因为三维运动通常会产生空间变化的光流场。
 - e.g. 在纯旋转的物体中, 旋转中心没有流动, 随着点到旋转中心的距离增加, 流矢量的大小也增加。
- 运动分割需要基于运动场的一些参数描述

二维光流估计与分割

- 一个真实的场景通常包含多个动作
- 不能跨运动边界施加平滑约束





运动分割方法

- 基于运动模型的分割
 - 改进的Hough变换
 - 参数聚类方法
 - 贝叶斯分割
 - 主导运动估计法
 - 同时估计和分割



总结

□ 基本原理

■ 与摄像机运动相对应的二维运动

- ✓ 投影变换, 仿射变换

■ 光流方程

- ✓ 由恒定亮度和小运动假设导出
- ✓ 运动估计的模糊性

■ 如何表示运动:

- ✓ 基于像素, 基于块, 基于区域, 全局表示, etc.

■ 估计标准:

- ✓ DFD (constant intensity)
- ✓ OF (constant intensity + small motion)
- ✓ Bayesian (MAP, DFD + motion smoothness)

■ 搜索方法:

- ✓ 穷举搜索, 梯度下降, 多分辨率



总结

- 一般方法:
 - 基于像素的运动估计
 - 基于块的运动估计
 - ✓ EBMA, 整数精度 vs. 分数精度, 快速算法
- 更先进的方法
 - 多分辨率方法
 - ✓ 避免局部极小值, 平滑运动区域, 减小计算量
 - 相位相关方法
 - 全局运动估计
 - ✓ 估计相机运动
 - 基于区域的运动估计
 - ✓ 更加合理的方法: 允许每个子对象区域有不同的运动
 - 运动分割