第十一章：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 (lill@ustc.edu.cn)
胡　洋 (eeyhu@ustc.edu.cn)

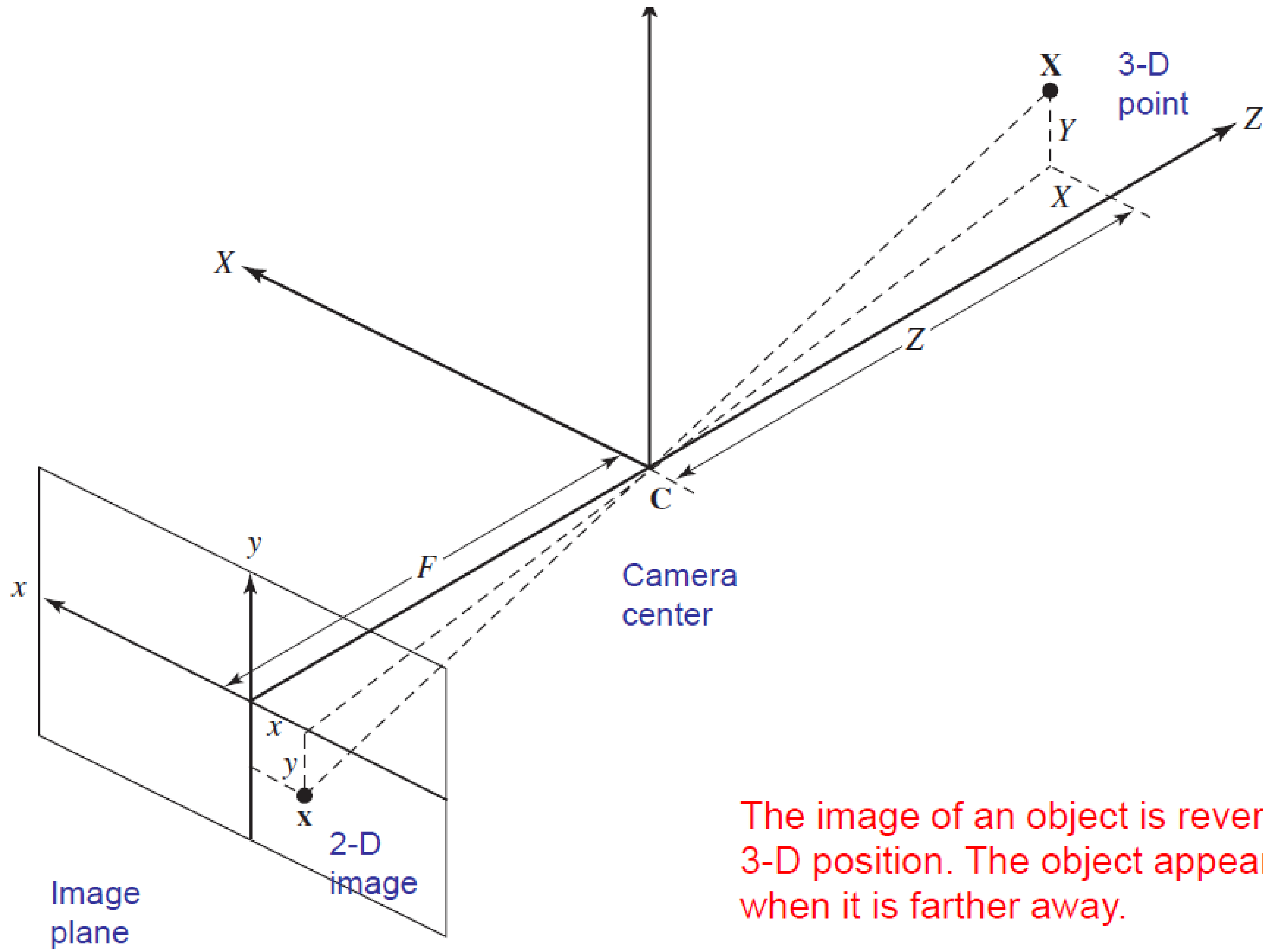
运动分析

- 二维运动模型
- 二维运动 vs. 光流
- 运动估计中的一般方法
- 基于块的运动估计
- 可形变块匹配
- 多分辨率运动估计
- 相位相关法
- 基于深度学习的运动分析

二维运动模型

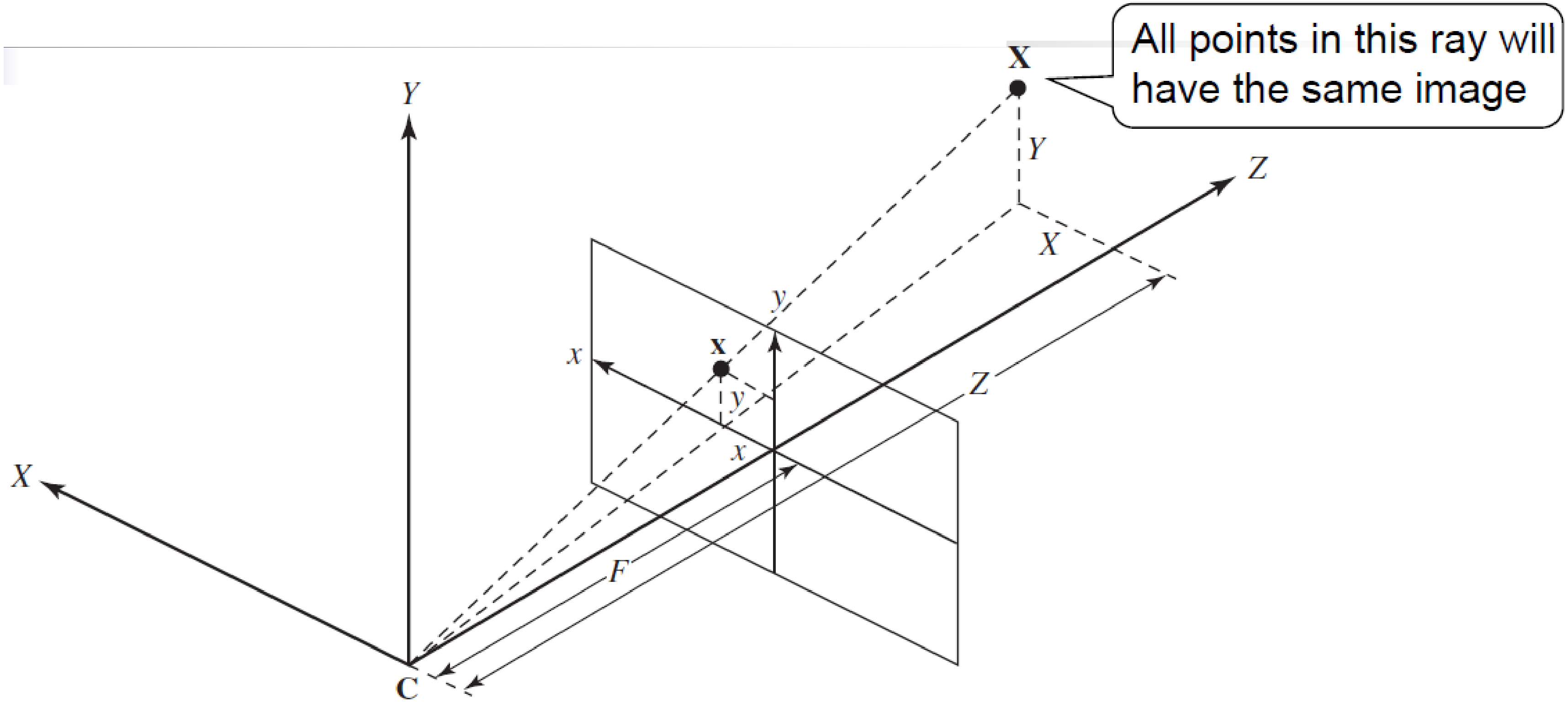
- 相机投影
- 三维运动
- 三维运动的投影
- 刚体目标的二维运动
  - 投影映射
- 投影映射的近似
  - 仿射模型
  - 双线性模型

针孔相机模型



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

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

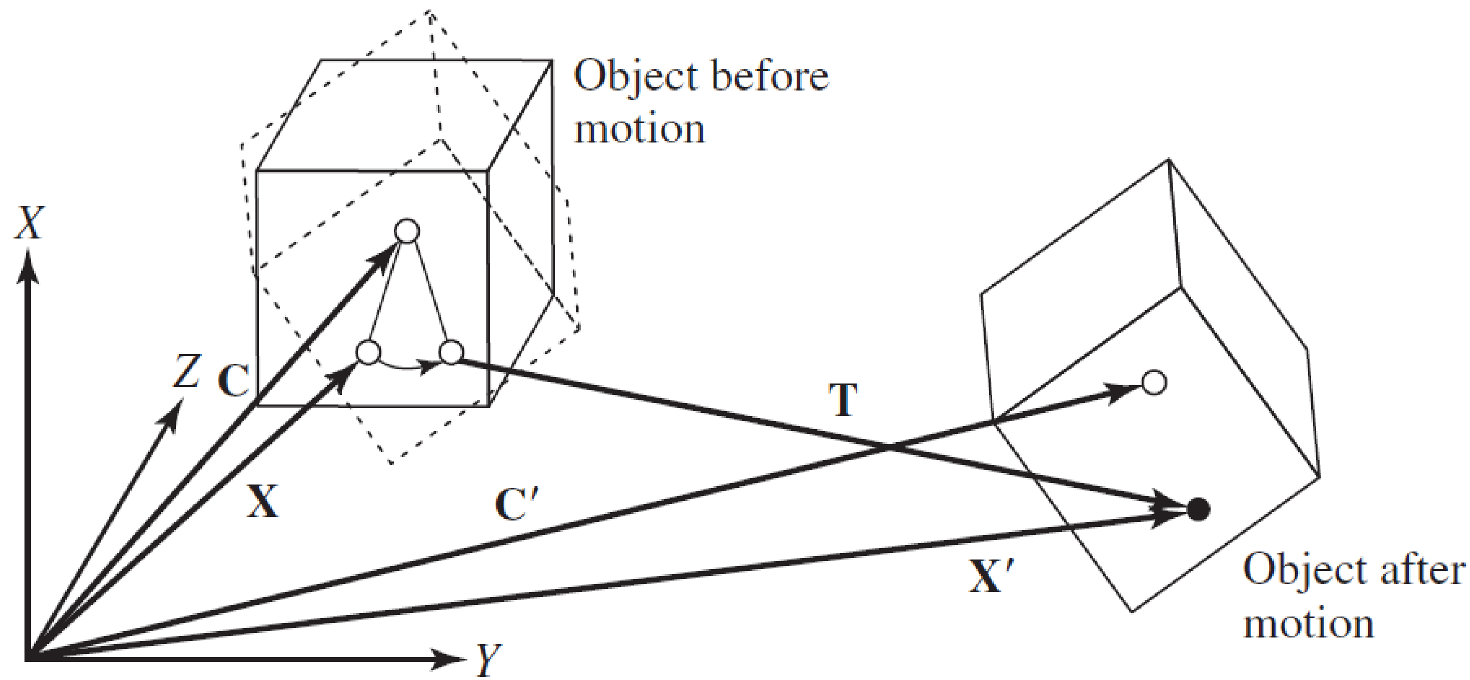
$$[R_x] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix}$$

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

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

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

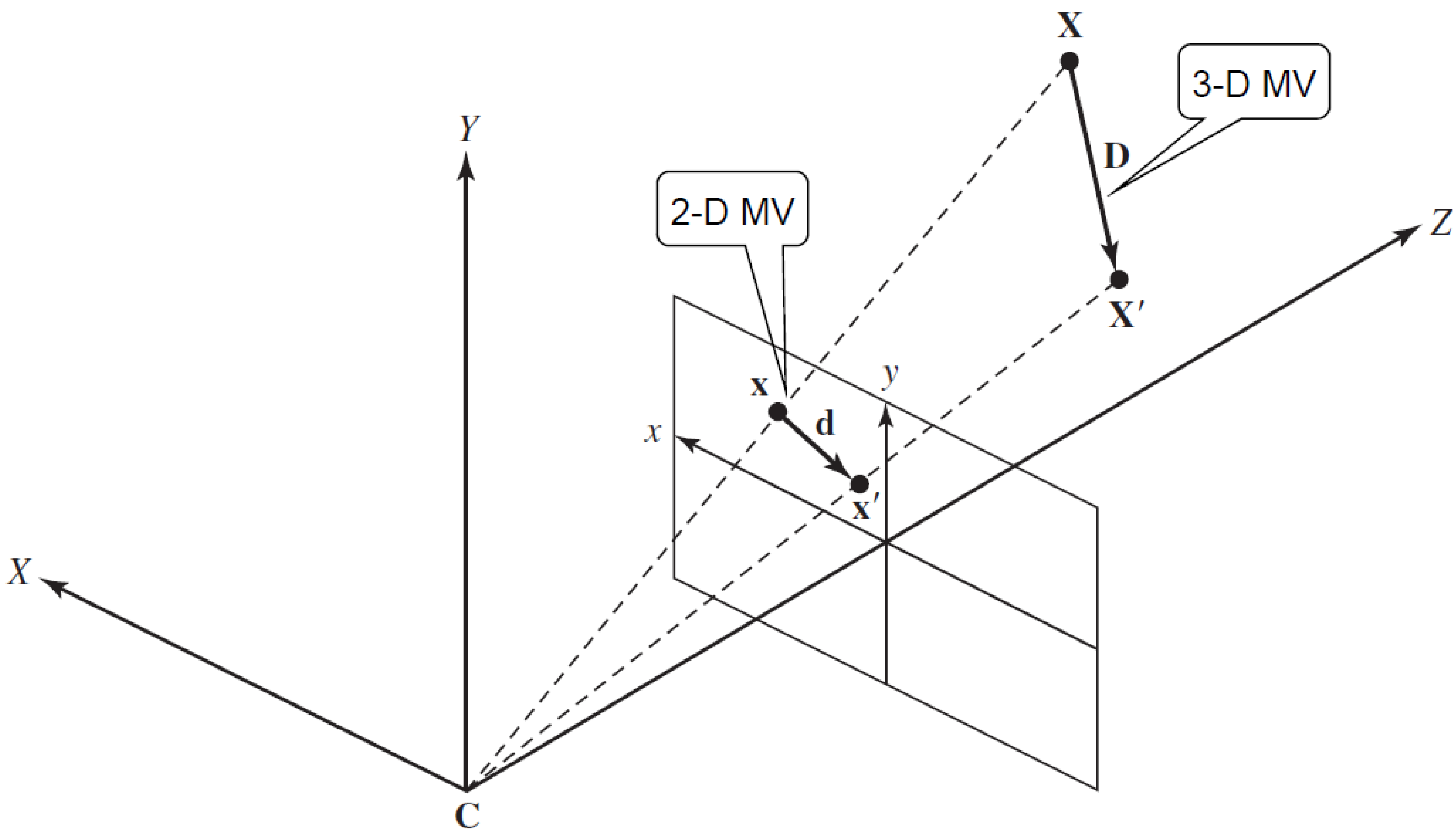
三维运动——刚体运动模型



Rotation and translation wrt. the object center ：

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

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

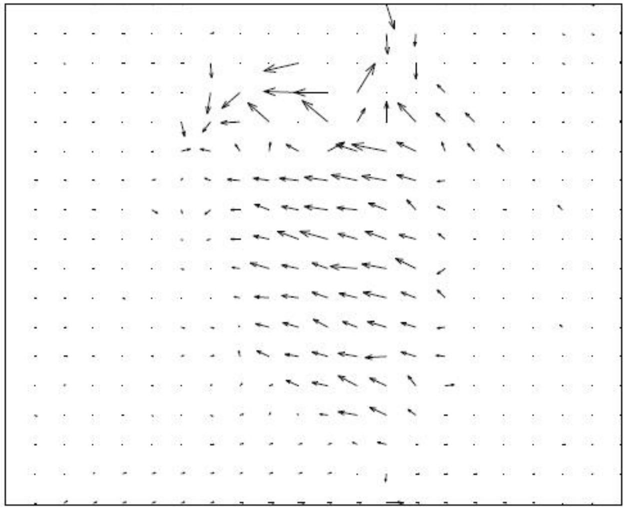
$$w(x;t_1,t_2)=x'$$

$$w(x)=x+d(x)$$

- 流矢量（速度矢量）

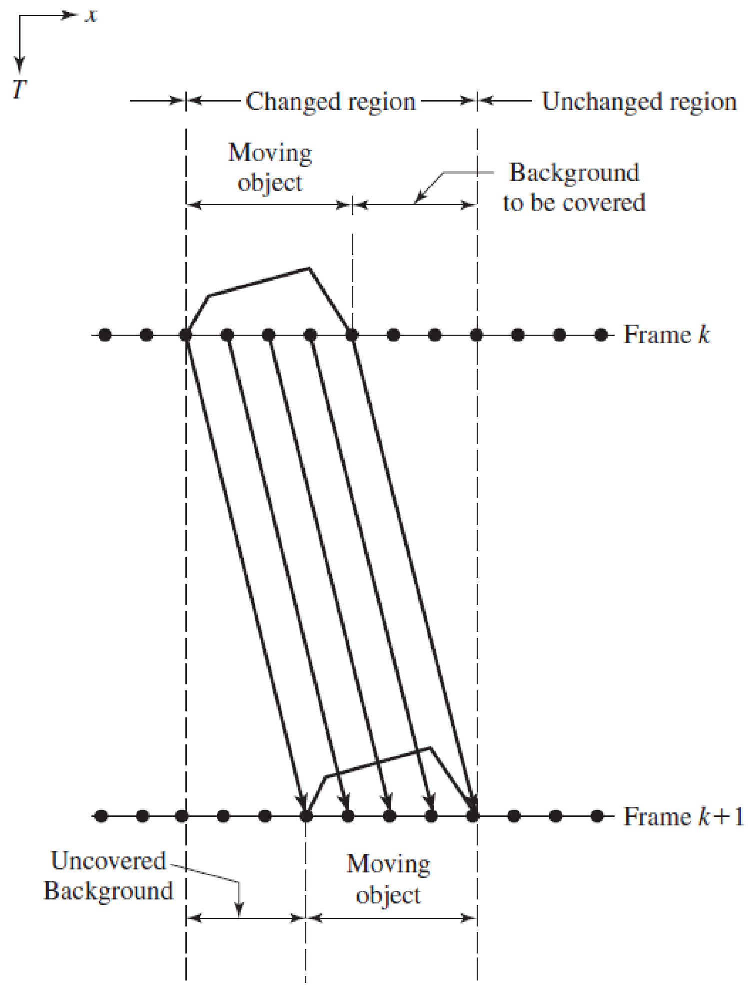
$$V=\frac{\partial d}{\partial t}=\left[\frac{\partial d_x}{\partial t},\frac{\partial d_y}{\partial t}\right]^T$$

一个典型的二维运动场



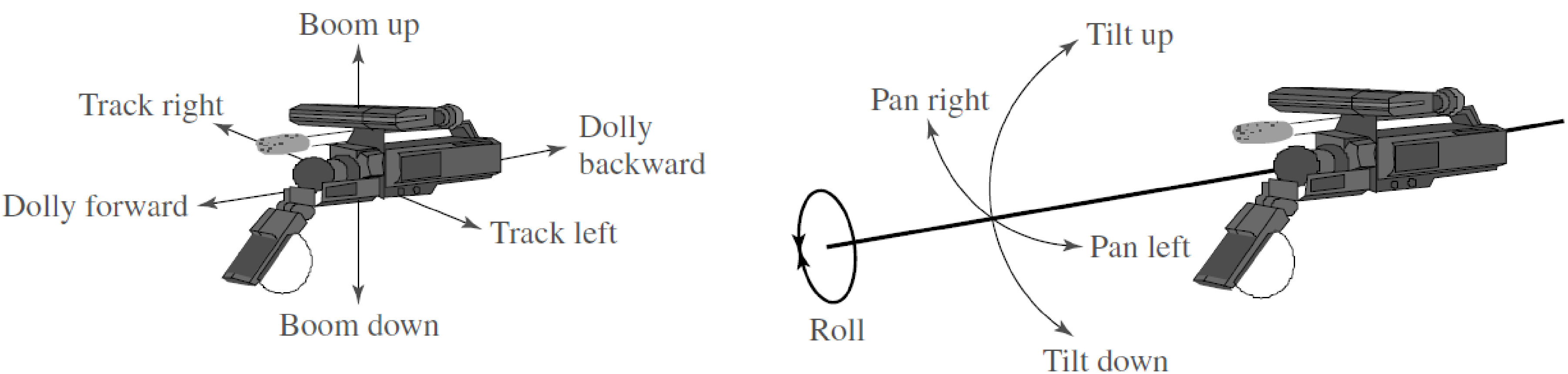
遮挡的影响

- 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)

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

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

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

相机推(Zoom)和滚(Roll)

- 推（zoom）：像平面与中心点距离（焦距）被改变

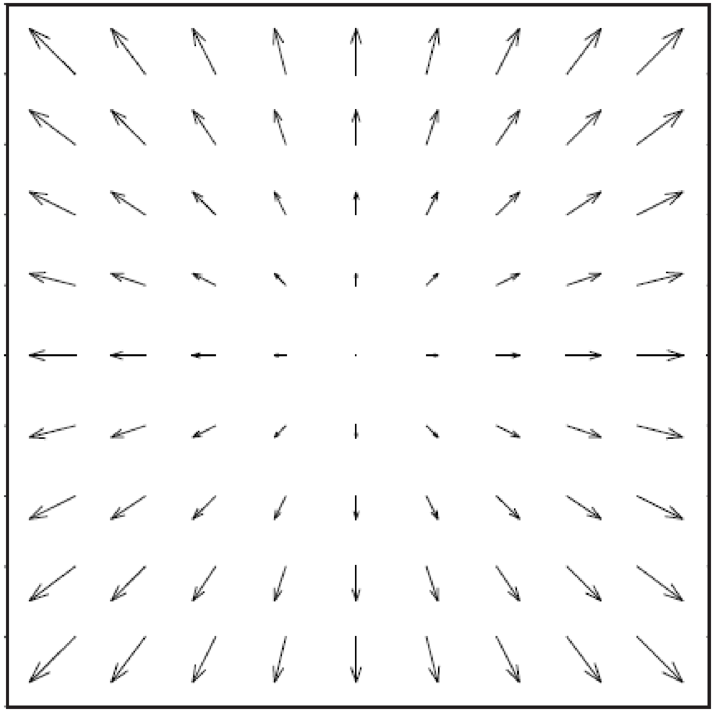
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \rho x \\ \rho y \end{bmatrix} \quad \Rightarrow \begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \begin{bmatrix} (1-\rho)x \\ (1-\rho)y \end{bmatrix} \quad (\rho = F'/F)$$

- 滚（roll）

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

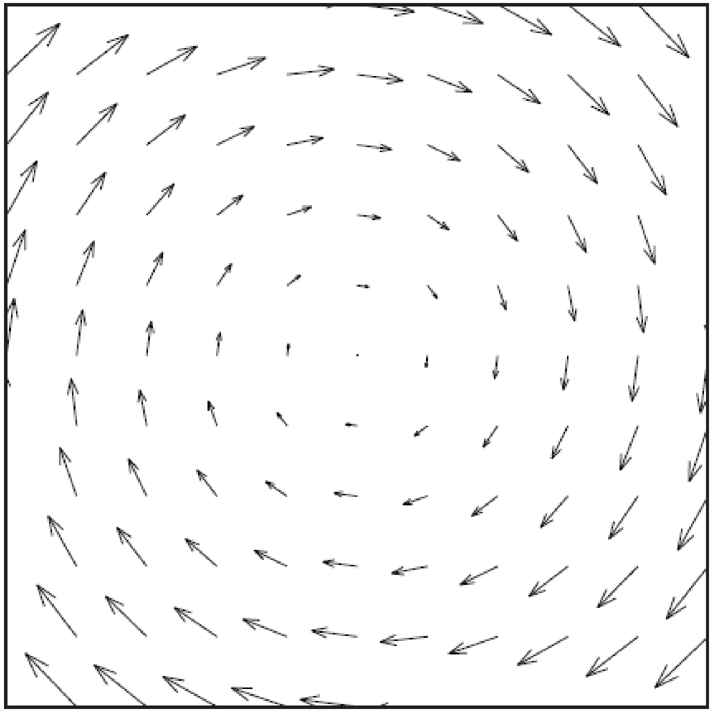
$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \begin{bmatrix} -\theta_z y \\ \theta_z x \end{bmatrix}$$

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：

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

$$= \begin{bmatrix} c_1 & -c_2 \\ c_2 & c_1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c_3 \\ c_4 \end{bmatrix}$$

- 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：

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

$$\xrightarrow[\text{Perspective Projection}]{}$$

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

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

投影映射

- 投影映射
  - 在Z方向没有平移运动
  - 目标有平坦表面

$$x' = \frac{a_0 + a_1 x + a_2 y}{1 + c_1 x + c_2 y}, \quad y' = \frac{b_0 + b_1 x + b_2 y}{1 + c_1 x + c_2 y}$$

- 实际图像可以分成多个包含平坦平面的区域

仿射和双线性模型

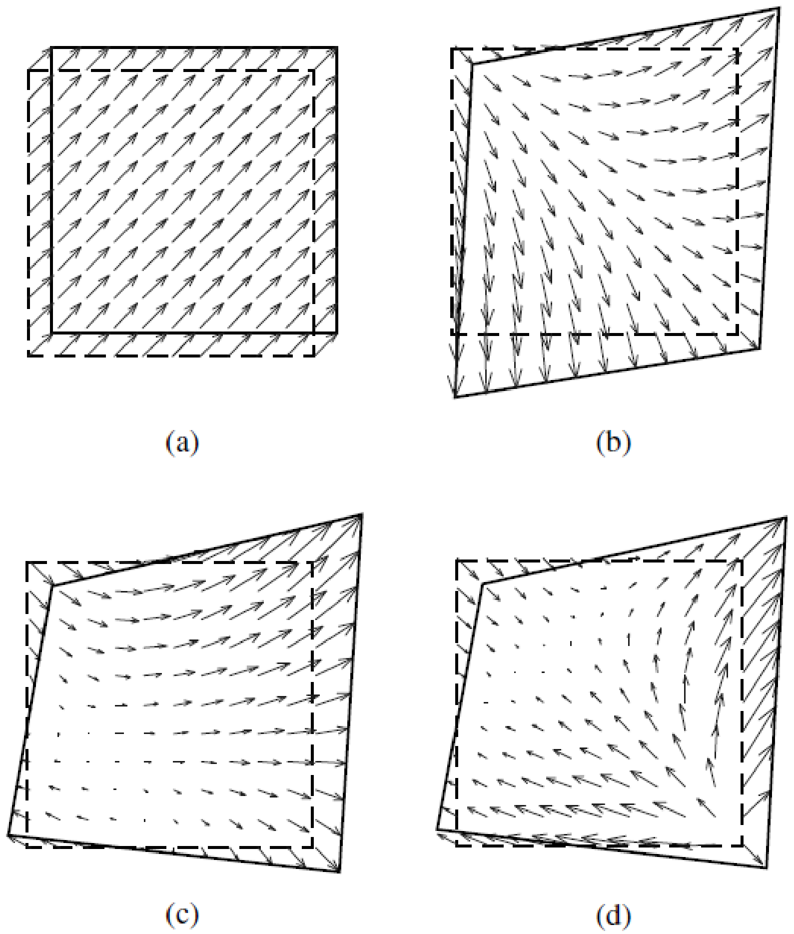
- 仿射 (6个参数):
  - 适合将一个三角形映射到一个三角形

$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x + a_2 y \\ b_0 + b_1 x + b_2 y \end{bmatrix}$$

- 双线性 (8个参数):
  - 适合将一个四边形映射为一个曲边四边形

$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x + a_2 y + a_3 xy \\ b_0 + b_1 x + b_2 y + b_3 xy \end{bmatrix}$$

不同二维运动模型的运动场



平移　(a)　(b)　仿射

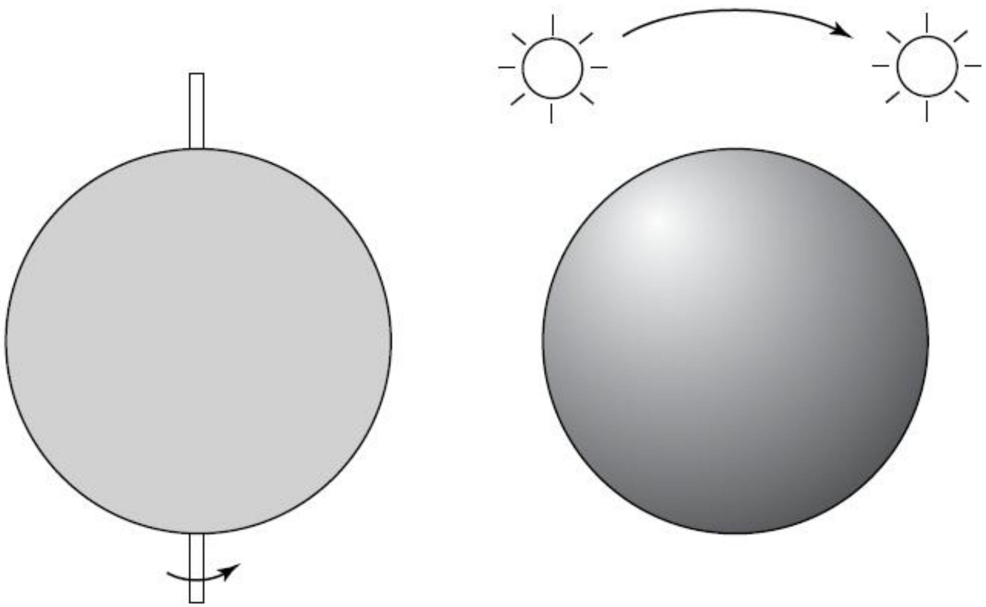
双线性　(c)　(d)　透视

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
- 基于块的运动估计
- 可形变块匹配
- 多分辨率运动估计
- 相位相关法
- 基于深度学习的运动分析

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:

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

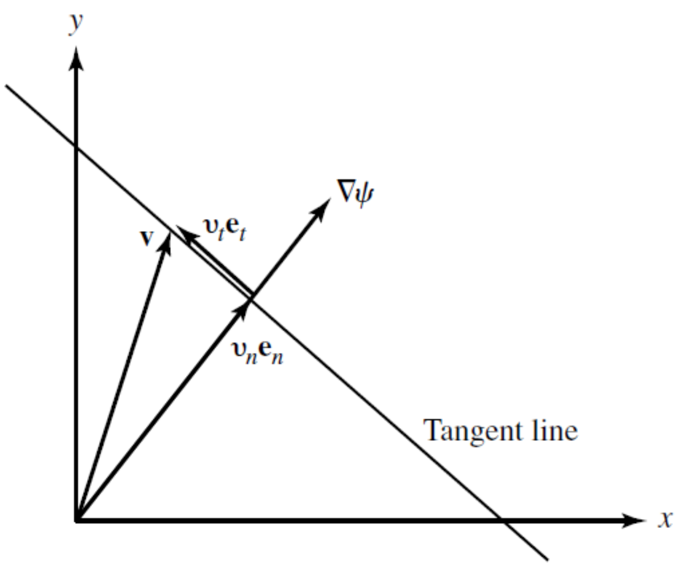
如何使用光流方程

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

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

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



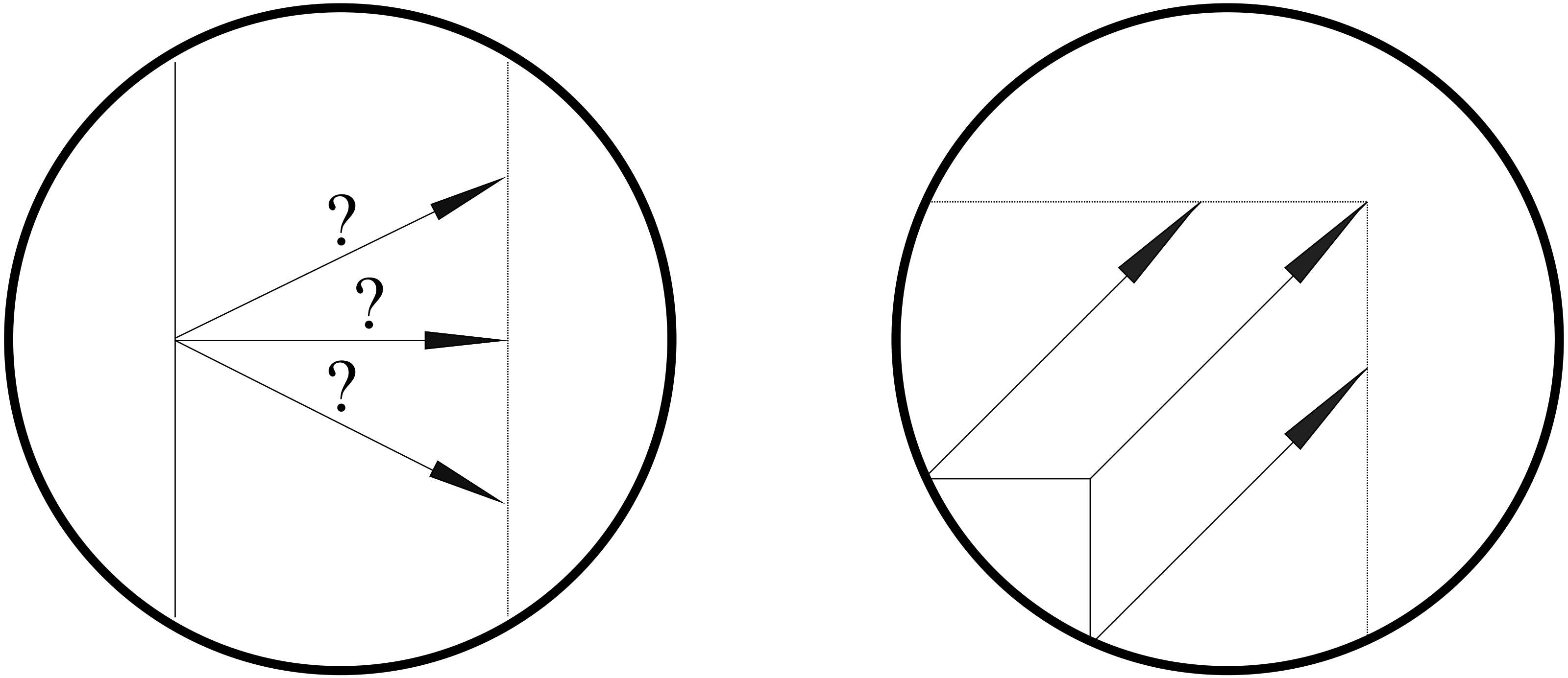
$$\nabla\psi^T\mathbf{v}+\frac{\partial\psi}{\partial t}=0$$

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

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

孔径问题 （Aperture problem）

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



运动分析

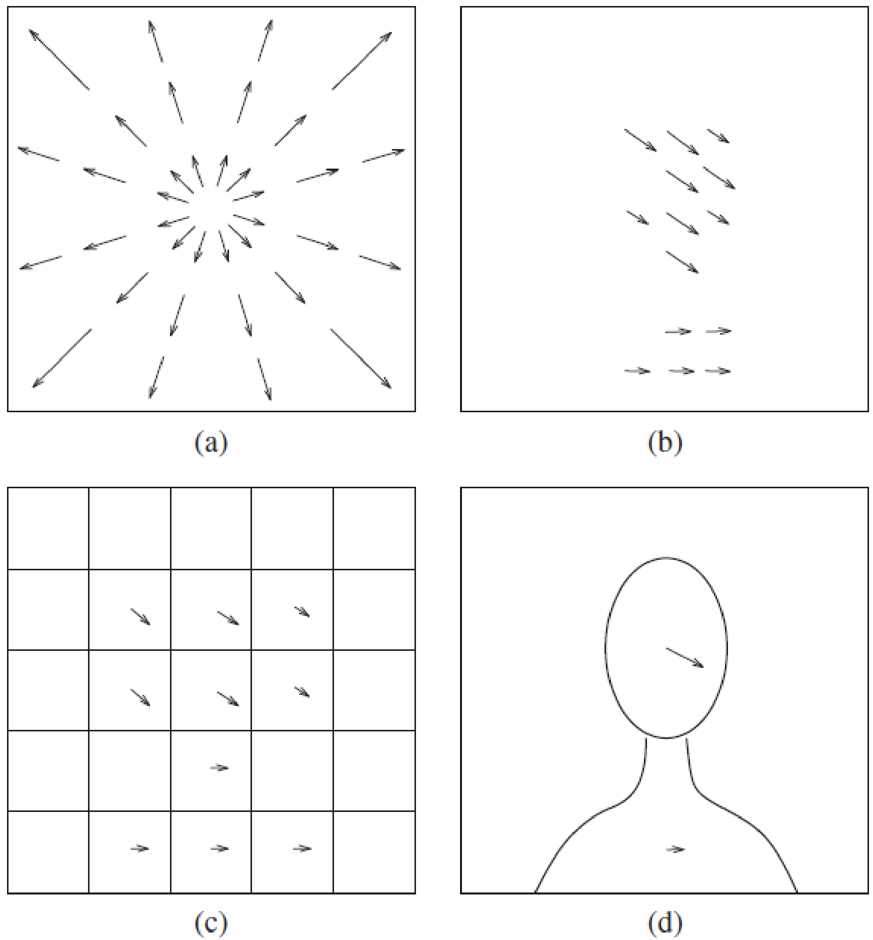
- 二维运动模型
- 二维运动 vs. 光流
- **运动估计中的一般方法**
- 基于块的运动估计
- 可形变块匹配
- 多分辨率运动估计
- 相位相关法
- 基于深度学习的运动分析

运动估计的一般考虑

- 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。



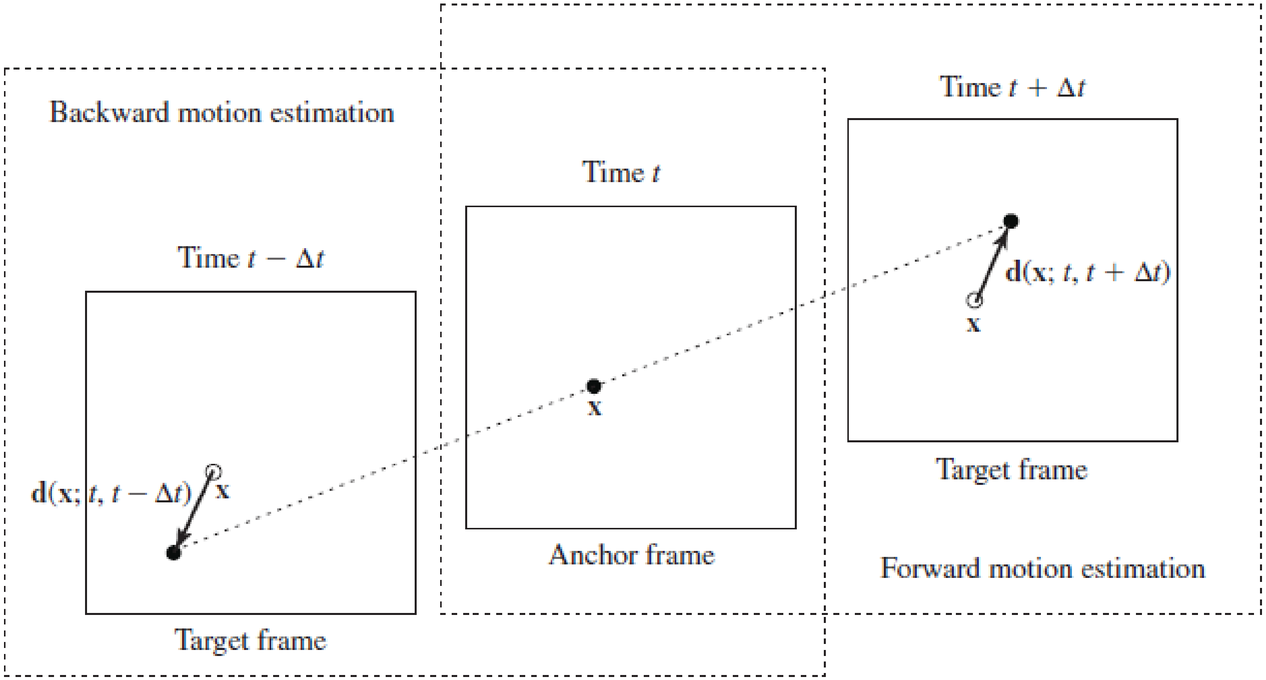
基于像素：
每一个像素有一个运动向量，在相邻运动向量之间有一些平滑约束。

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。

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）

运动估计准则符号定义



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)

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

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

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

- 基于光流方程准则 (OF criterion)

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

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

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

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)

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

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

- 贝叶斯准则 (Bayesian criterion)：最大化后验概率

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

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

$$\mathbf{d}_{\mathrm{MAP}} = \operatorname{argmax}_{\mathbf{d}} \{ P(\Psi = \psi_2 | \mathcal{D} = \mathbf{d}; \psi_1) P(\mathcal{D} = \mathbf{d}; \psi_1) \}$$

$$\begin{aligned} \mathbf{d}_{\mathrm{MAP}} &= \operatorname{argmax}_{\mathbf{d}} \{ P(\mathcal{E} = e) P(\mathcal{D} = \mathbf{d}; \psi_1) \} \\ &= \operatorname{argmin}_{\mathbf{d}} \{ -\log P(\mathcal{E} = e) - \log P(\mathcal{D} = \mathbf{d}; \psi_1) \} \end{aligned}$$

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，通常可以得到闭式解
  - 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
- **基于块的运动估计**
- 可形变块匹配
- 多分辨率运动估计
- 相位相关法
- 基于深度学习的运动分析

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

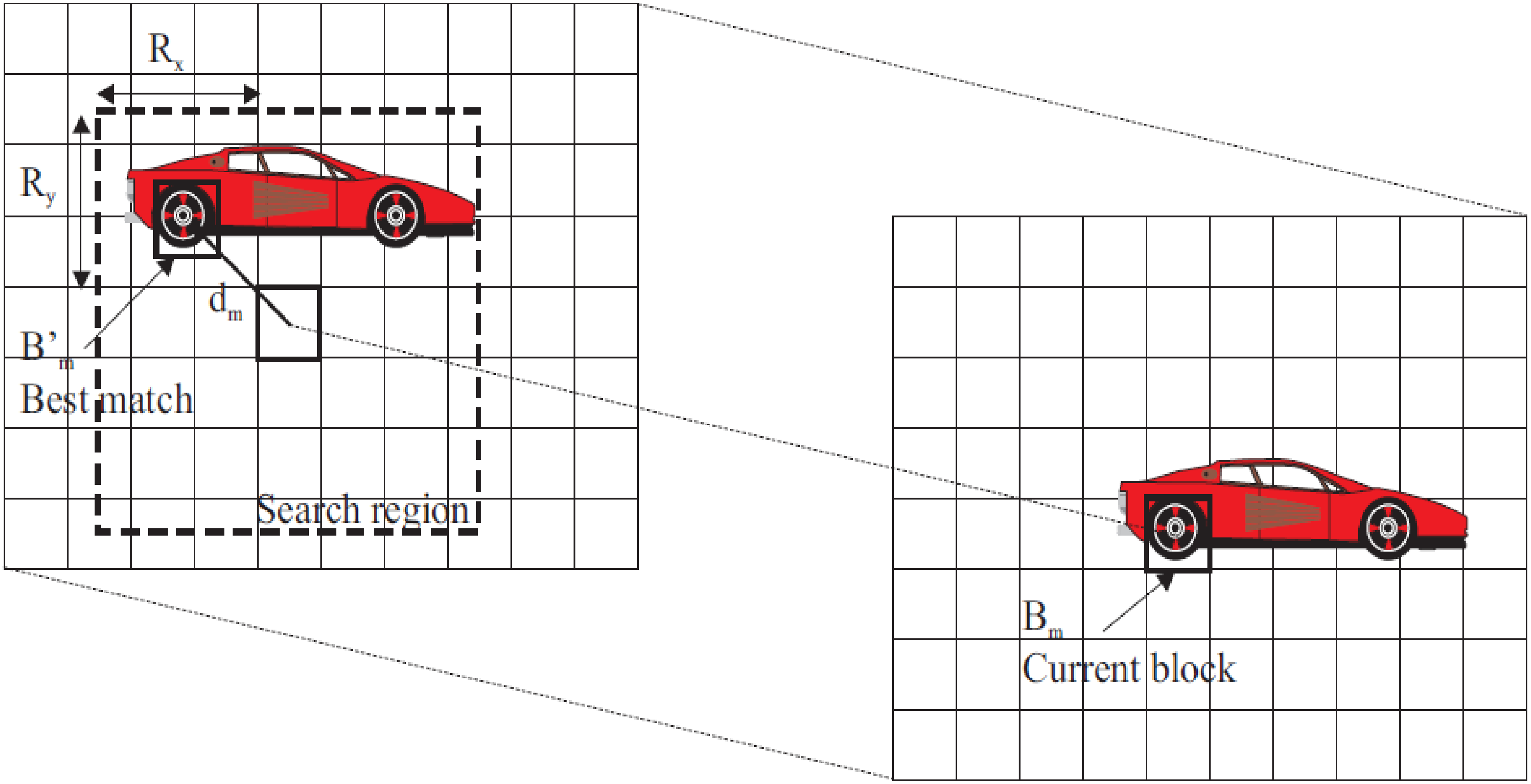
块匹配算法 (BMA)

- 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：

$$E_{\mathrm{DFD}}(\mathbf{d}_{\mathrm{m}}) = \sum_{\mathbf{x}\in B_m} \left|\psi_2(\mathbf{x}+\mathbf{d}_m) - \psi_1(\mathbf{x})\right|^p \to \min$$

- 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

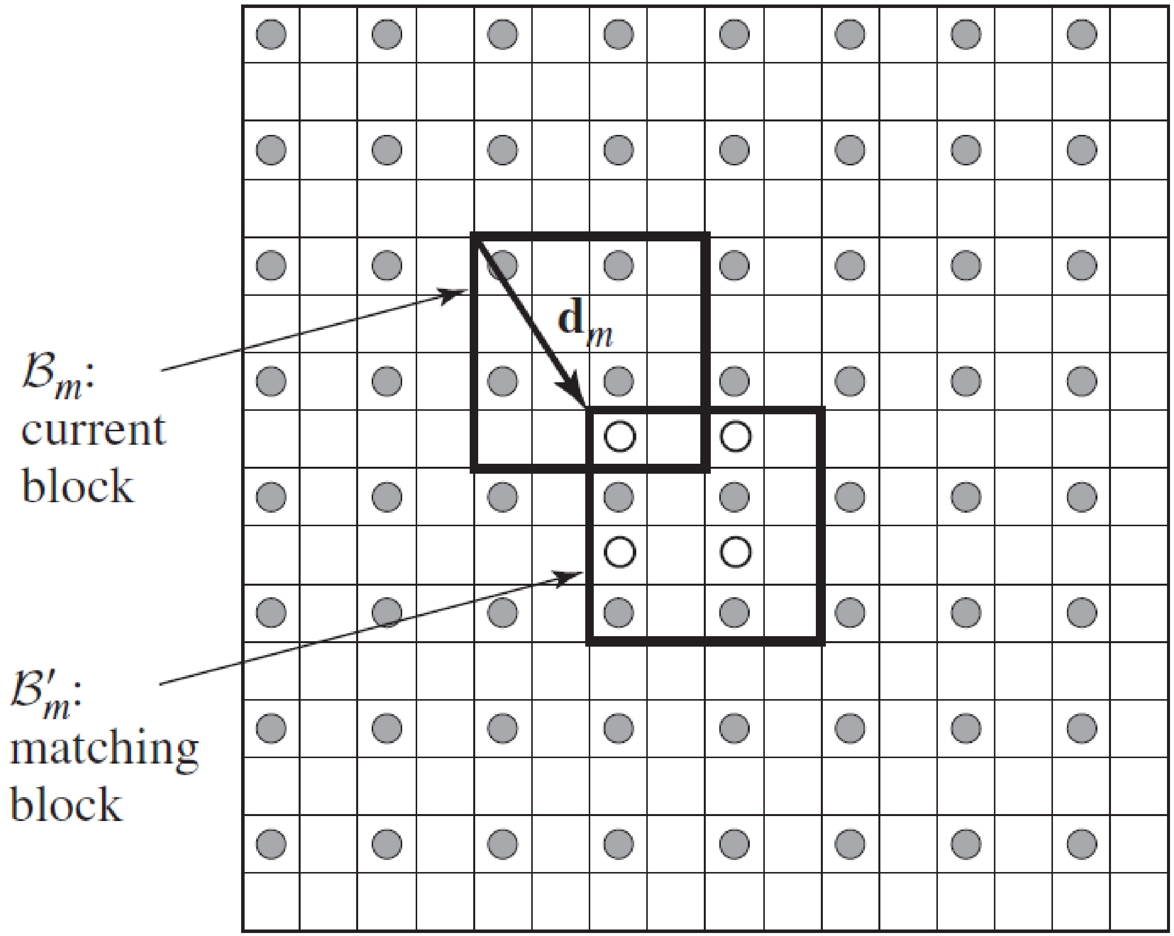
- 假设
  - 图像尺寸：$M \times M$
  - 块尺寸：$N \times N$
  - 搜索范围：$(-R, R)$in each dimension
  - 搜索步长：1 pixel (assuming integer MV)
- 操作数 (Operation counts)：

  (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: $(M/N)^2(2R+1)^2N^2 = M^2(2R+1)^2$
    - ✓ 独立于块尺寸！
- 例子：M=512, N=16, R=16, 30 fps
  - 总操作数 = 2.85x10^8/frame*30 frame/s =8.55x10^9/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

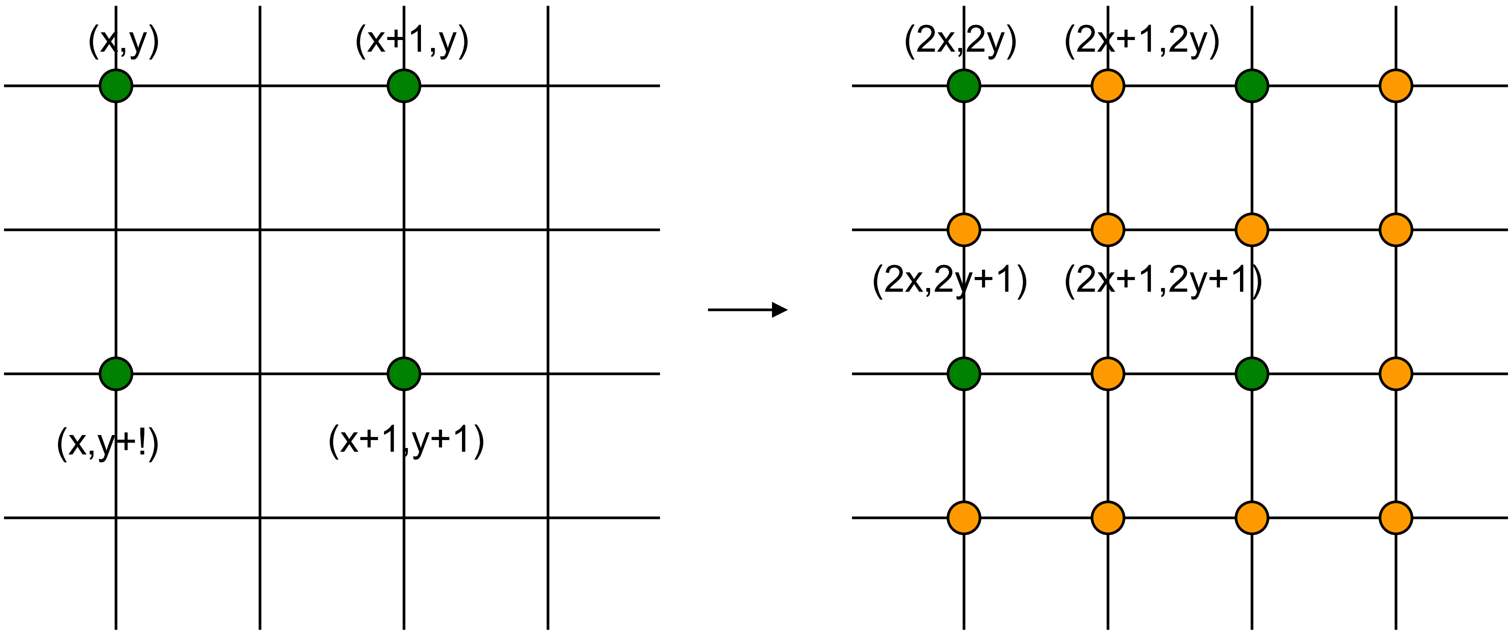
- MV估计中，搜索步长并不一定是一个整数，在实际情况下，分数步长可能更合适
- 半像素精度EBMA：step-size=1/2 pixel in both dimension
- 困难：
  - 目标帧仅有整数像素点
- 解决方案：
  - 在搜索之前目标帧先进行2倍内插
- 计算复杂度：
  - 4倍于整数像素精度，并加上额外的插值开销
- 快速算法：
  - 首先以整数精度进行搜索，然后在小范围内以半像素精度进行细化

半像素精度EBMA



$\mathbf{d}_m$
$\mathcal{B}_m$: current block
$\mathcal{B}'_m$: matching block

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

anchor frame

Motion field

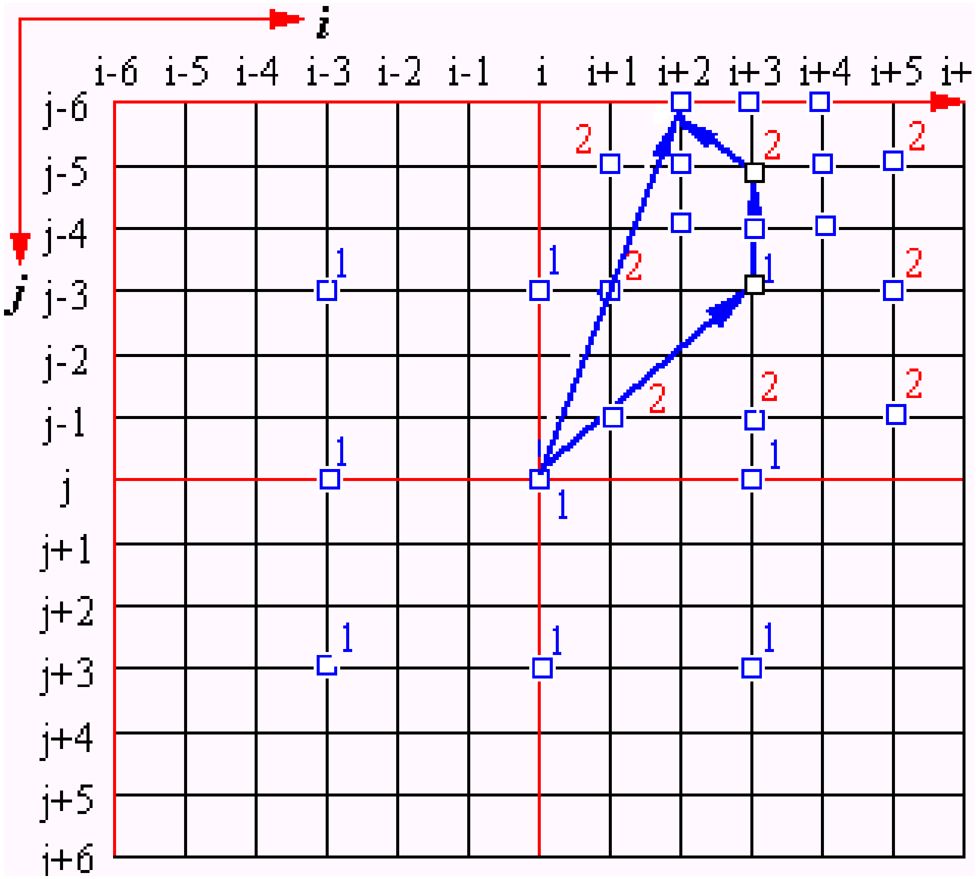
Predicted anchor frame (29.86dB)

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

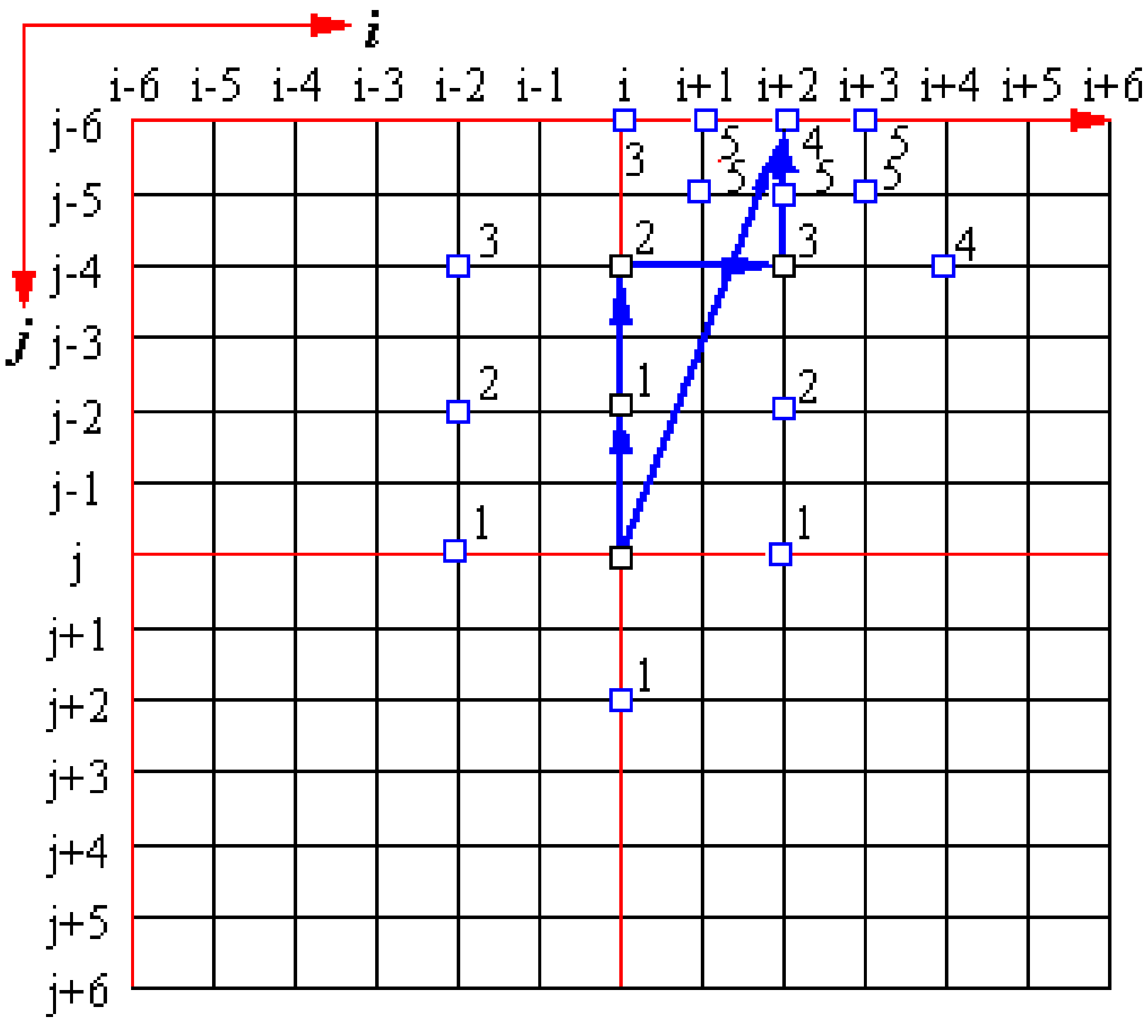
For example

R=32

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

3Step:41 = 8*5+1

二维对数搜索法



i
j
i-6 i-5 i-4 i-3 i-2 i-1 i i+1 i+2 i+3 i+4 i+5 i+6
j-6
j-5
j-4
j-3
j-2
j-1
j
j+1
j+2
j+3
j+4
j+5
j+6
1
1
1
2
2
1
3
2
3
3
4
4
5
5
5
5
5

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

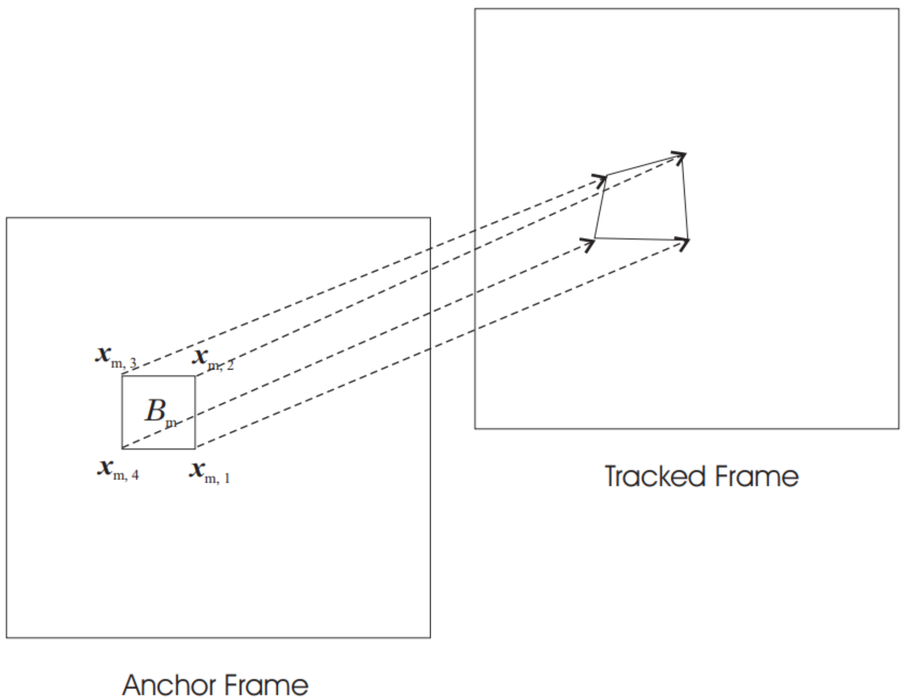
运动分析

- 二维运动模型
- 二维运动 vs. 光流
- 运动估计中的一般方法
- 基于块的运动估计
- **可形变块匹配**
- 多分辨率运动估计
- 相位相关法
- 基于深度学习的运动分析

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

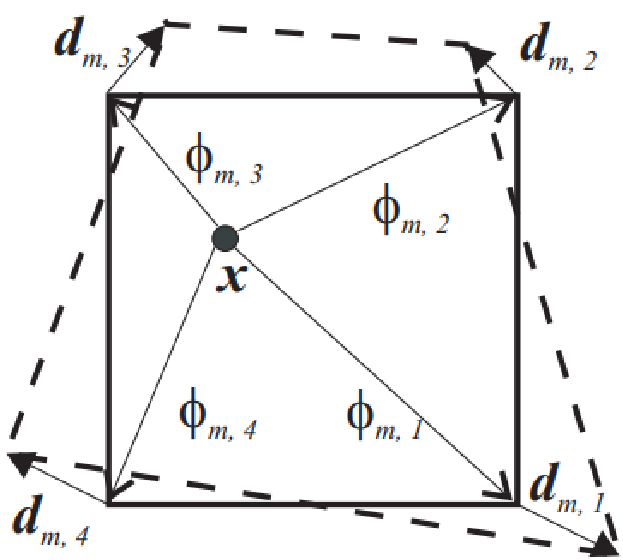
- 可形变块匹配算法
  - 面向更高阶运动的图像块匹配算法



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
  - ■ 使用顶点的运动信息表示

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



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

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

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

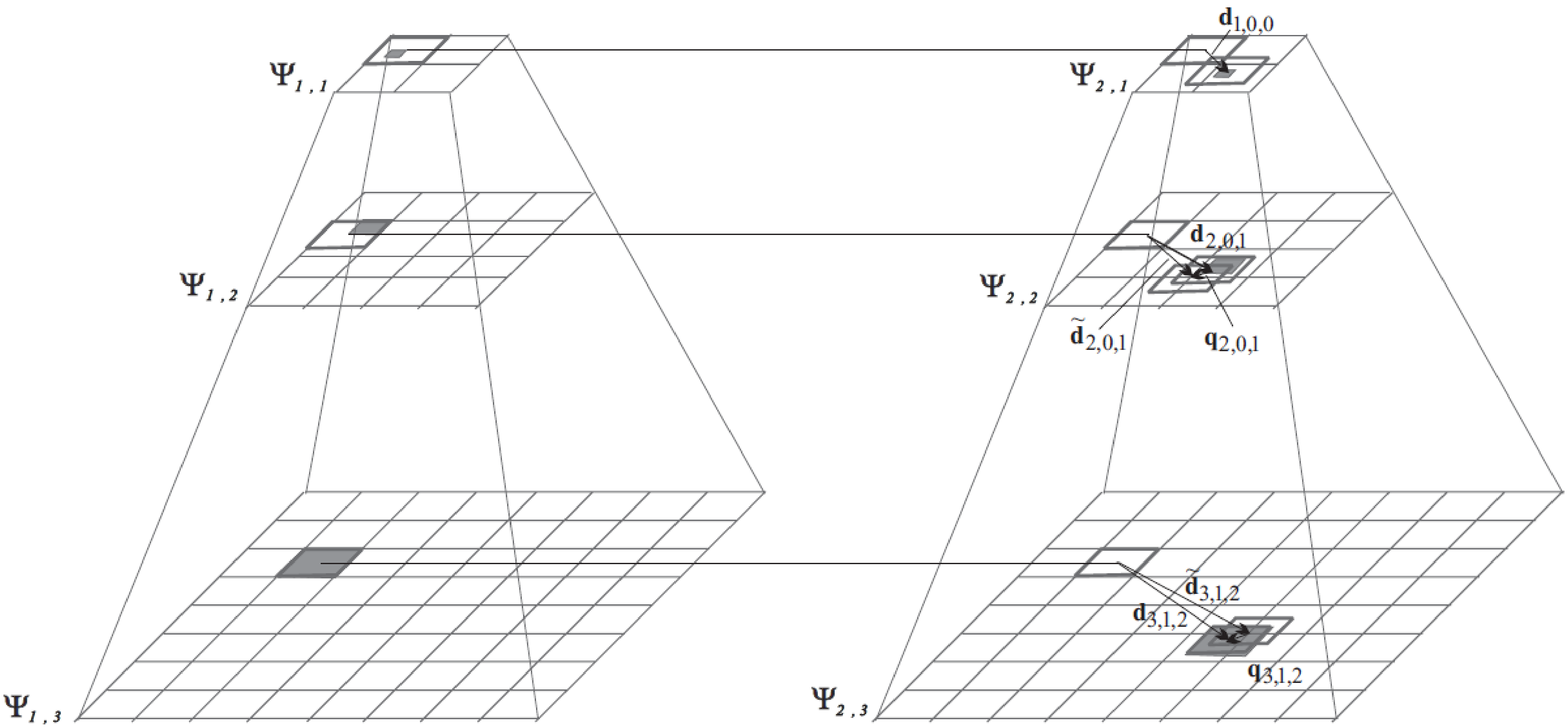
运动分析

- 二维运动模型
- 二维运动 vs. 光流
- 运动估计中的一般方法
- 基于块的运动估计
- 可形变块匹配
- **多分辨率运动估计**
- 相位相关法
- 基于深度学习的运动分析

多分辨率运动估计

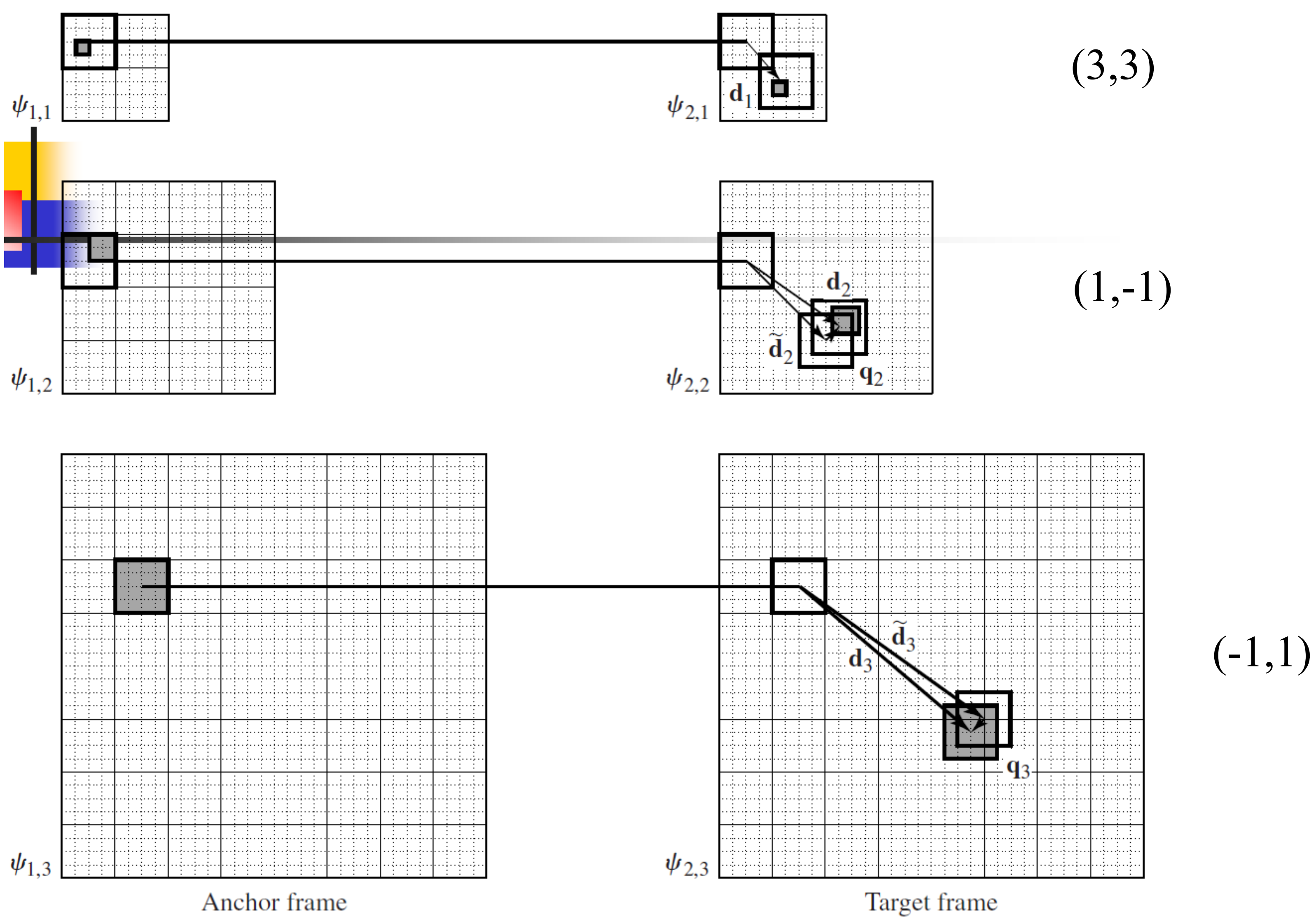
- BMA存在缺陷
  - 除非使用穷举搜索，否则可能难以达到全局最优解
  - 穷举搜索需要非常大的计算量
  - 基于块的平移运动模型并不总是合适的
- 多分辨率估计方法
  - 解决上述前两个问题
  - 首先在低通滤波、下采样的图像对上，进行低分辨率下的运动估计
    - ✓ 通常能得到一个接近于真实运动场的解
  - 然后在较小的搜索范围内以更高的分辨率逐步改善初始解
    - ✓ 降低计算量
  - 可以应用于不同的运动场景下，后续内容中我们只集中介绍其在BMA中的应用

多分辨率运动估计



$\Psi_{1,1}$
$\Psi_{1,2}$
$\Psi_{1,3}$
$\mathbf{d}_{1,0,0}$
$\Psi_{2,1}$
$\mathbf{d}_{2,0,1}$
$\Psi_{2,2}$
$\tilde{\mathbf{d}}_{2,0,1}$
$\mathbf{q}_{2,0,1}$
$\tilde{\mathbf{d}}_{3,1,2}$
$\mathbf{d}_{3,1,2}$
$\mathbf{q}_{3,1,2}$
$\Psi_{2,3}$

分层块匹配算法 (HBMA)



分层块匹配算法 (HBMA)

Number of levels: L

lth 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: $d_l(X) = \tilde{d}_l(X) + q_l(X)$

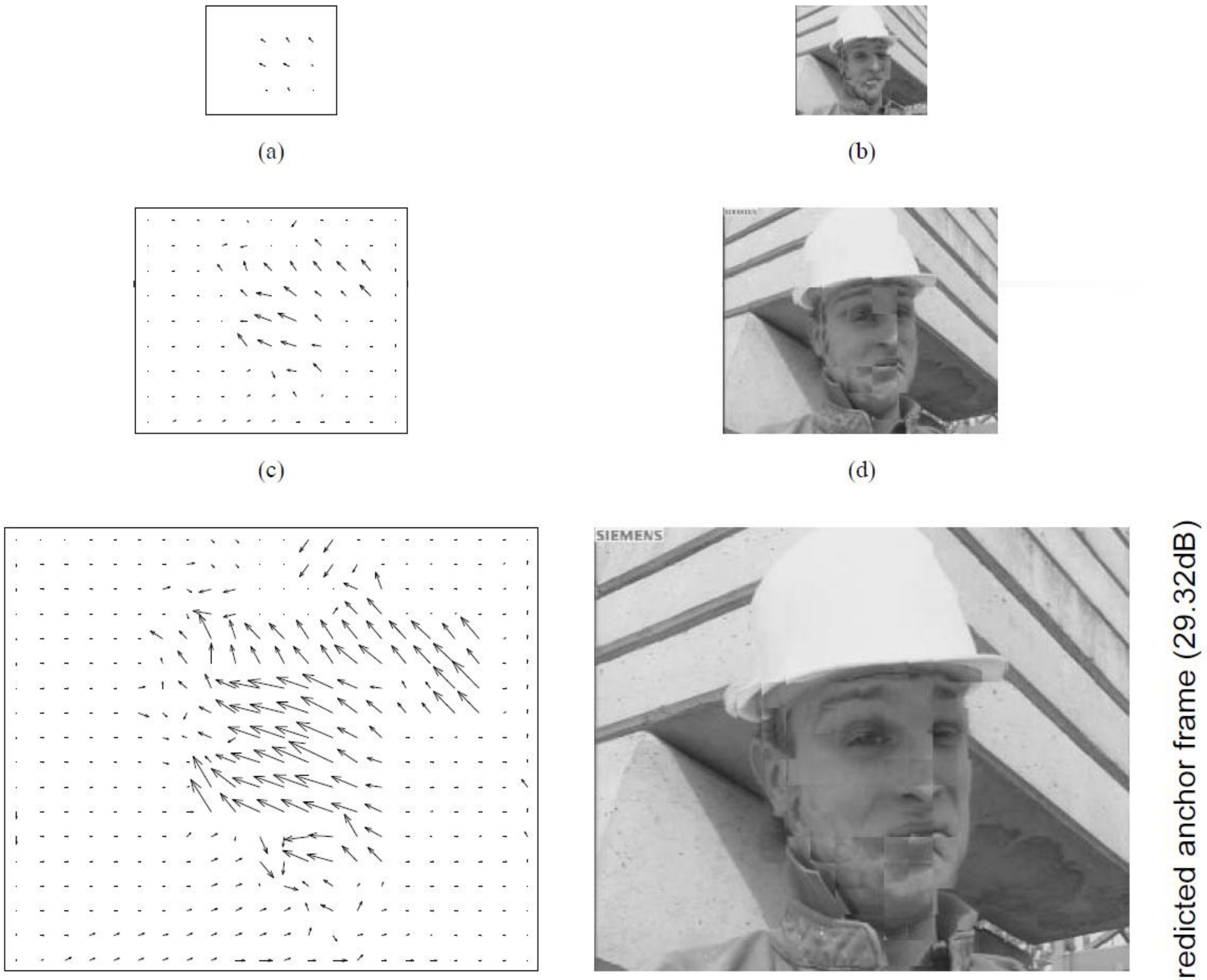
MV at lth level prediction:

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

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)



(a) (b)

(c) (d)

(e) (f)

Example: Three-level HBMA

HBMA复杂度

- 假设
  - 图像尺寸：MxM
  - 块尺寸：NxN 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= MxM，块尺寸= NxN，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})

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

- HBMA总操作数

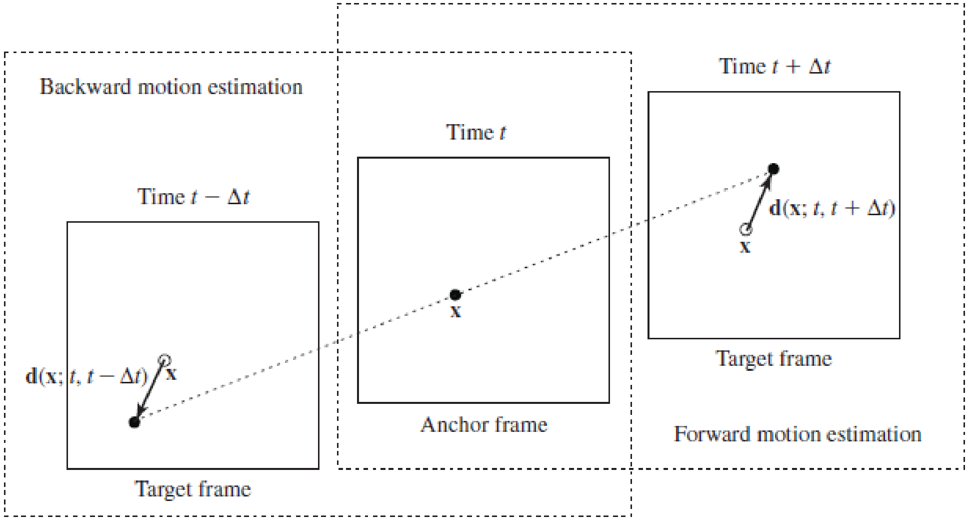
$$\sum_{l=1}^{L}\left(M / 2^{L-l}\right)^2\left(2R / 2^{L-1} + 1\right)^2 \approx \frac{1}{3} 4^{-(L-2)} 4M^2R^2$$

- 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
- 基于块的运动估计
- 可形变块匹配
- 多分辨率运动估计
- **相位相关法**
- 基于深度学习的运动分析

相位相关法



识别相位相关函数的峰值 (PCF)

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

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

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

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

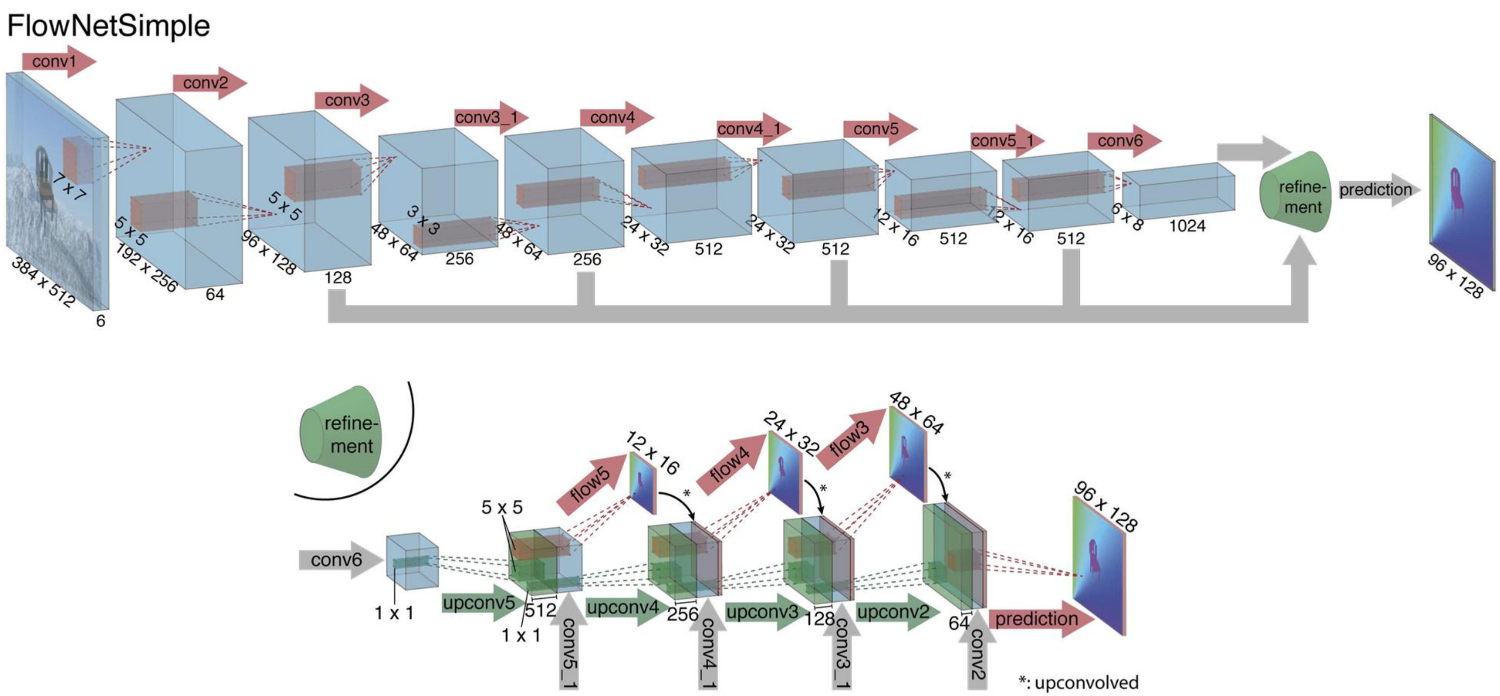
- Note
  - 减轻边界采样效应：空间域加权窗函数
  - 广泛应用于图像配准
  - 优点：对光照变化不敏感

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
- 基于块的运动估计
- 可形变块匹配
- 多分辨率运动估计
- 相位相关法
- **基于深度学习的运动分析**

基于图像对的光流估计

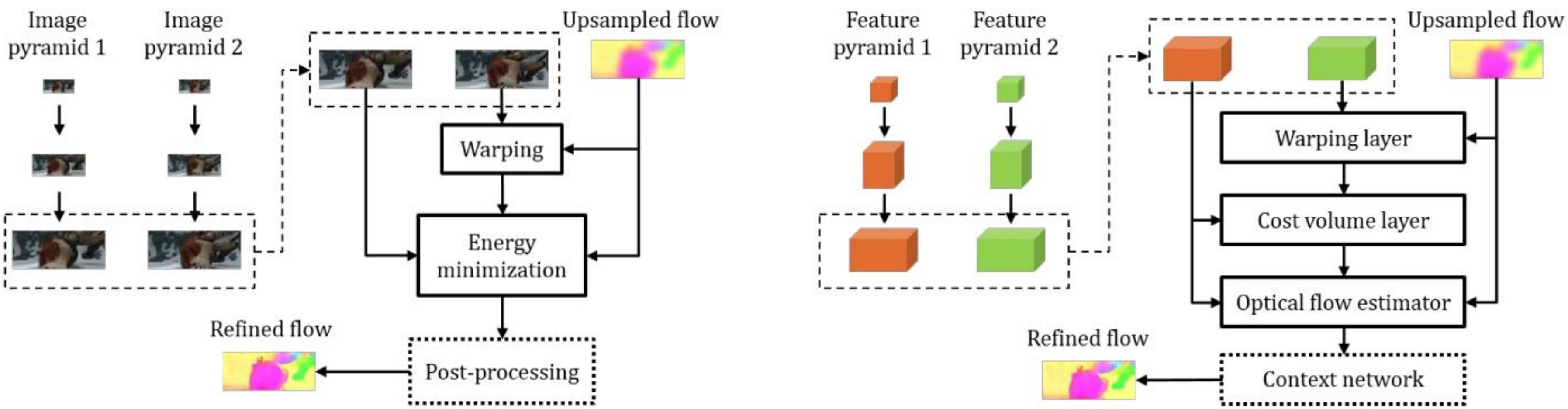
- FlowNet
  - 第一个基于深度学习的光流网络
  - 构建了仿真数据集FlyingChairs
  - 类似于U-Net的网络结构



[1] Philipp Fischer, et al. "FlowNet: Learning Optical Flow with Convolutional Networks." In ICCV 2015.

基于图像对的光流估计

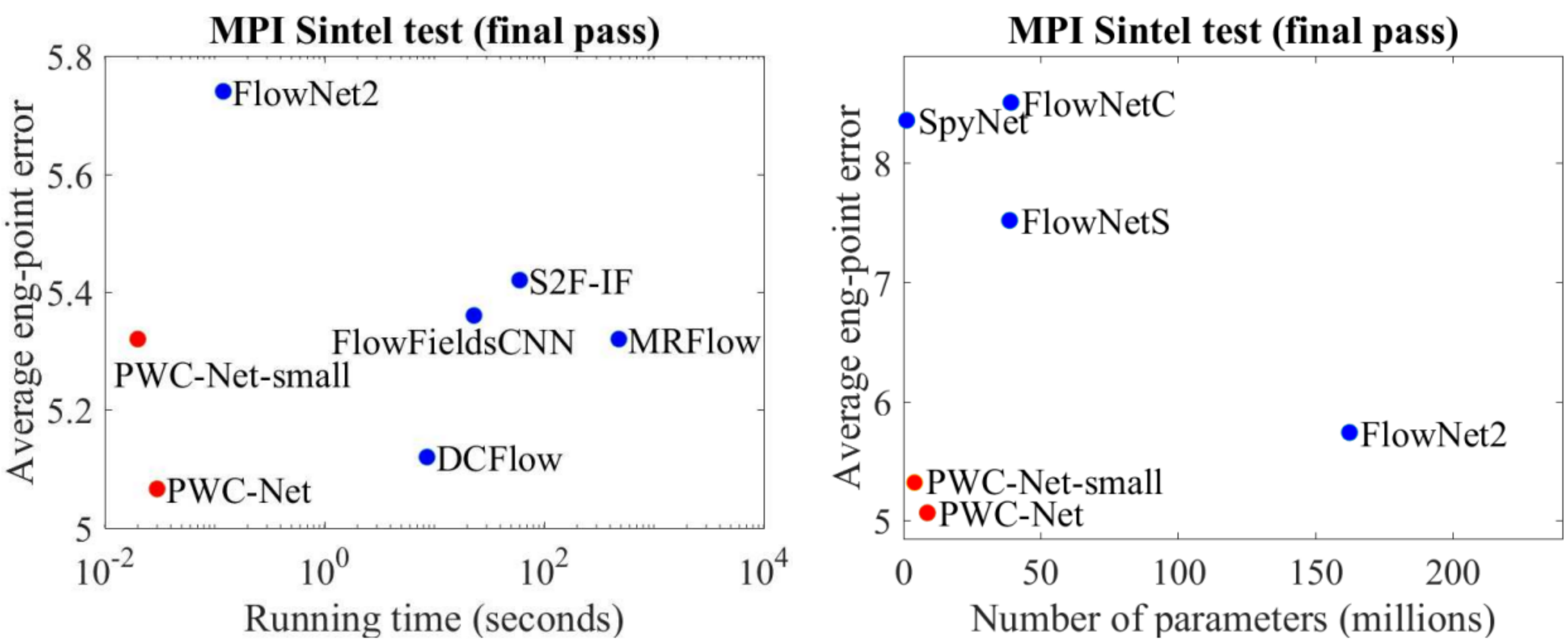
- PWC-Net
  - Feature pyramid
  - Cost volume layer: store the matching costs for associating a pixel with its corresponding pixels at the next frame
  - Context network: enlarge the receptive field size of each output unit at the desired pyramid level



[1] Deqing Sun, et al. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume." In CVPR 2018.

基于图像对的光流估计

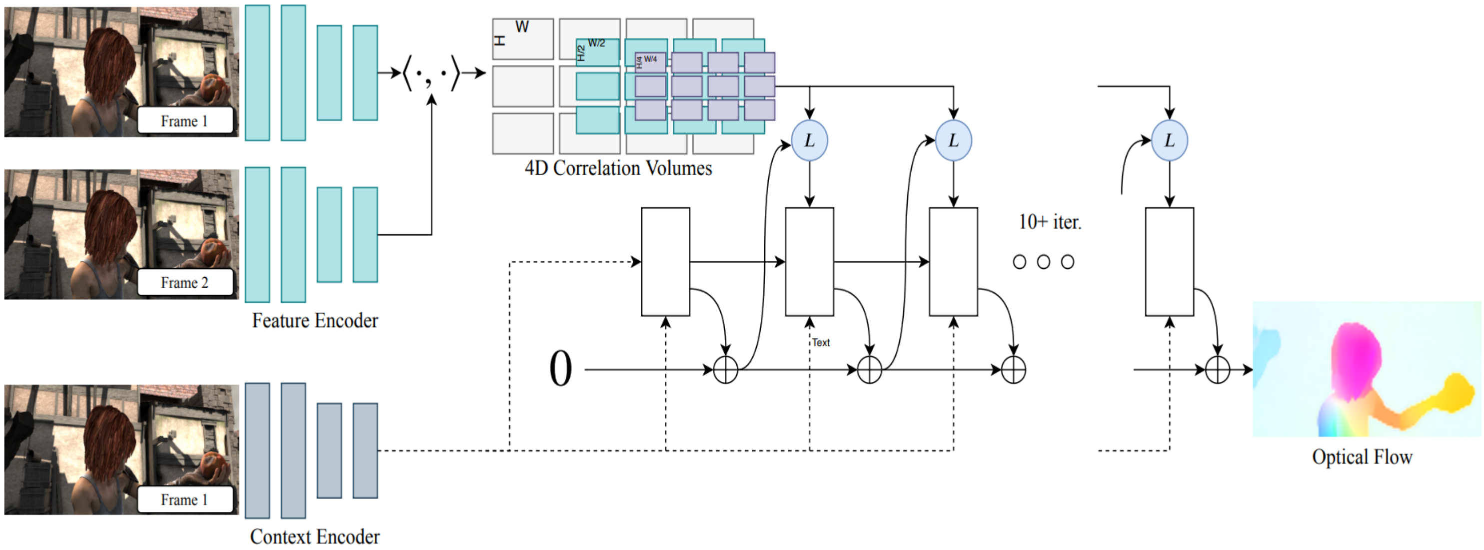
- PWC-Net
  - PWC achieves the best performance at that time
  - SpyNet has the lowest complexity



[1] Deqing Sun, et al. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume." In CVPR 2018.

基于图像对的光流估计

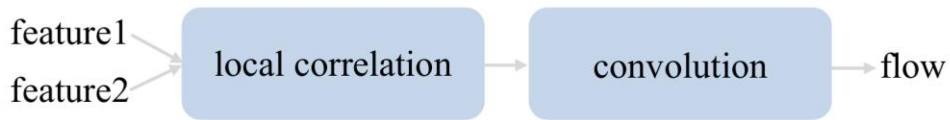
- RAFT
  - 高分辨率下直接计算所有像素对的cost volume；inference极端多次迭代仅需计算一次
  - 迭代网络是一个GRU，每次迭代共享参数，理论上可以无数次迭代；**较为普适的一种渐进式思路**
  - 取得了当时最优的性能和泛化能力



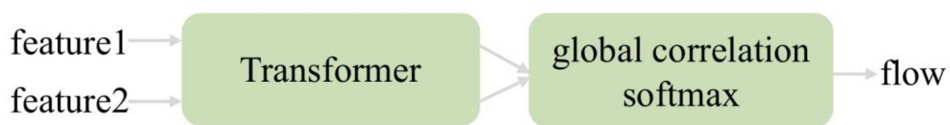
[1] Teed Zachary, et al. "RAFT: Recurrent All-Pairs Field Transforms for Optical Flow." In ECCV 2020. **(Best paper)**

基于图像对的光流估计

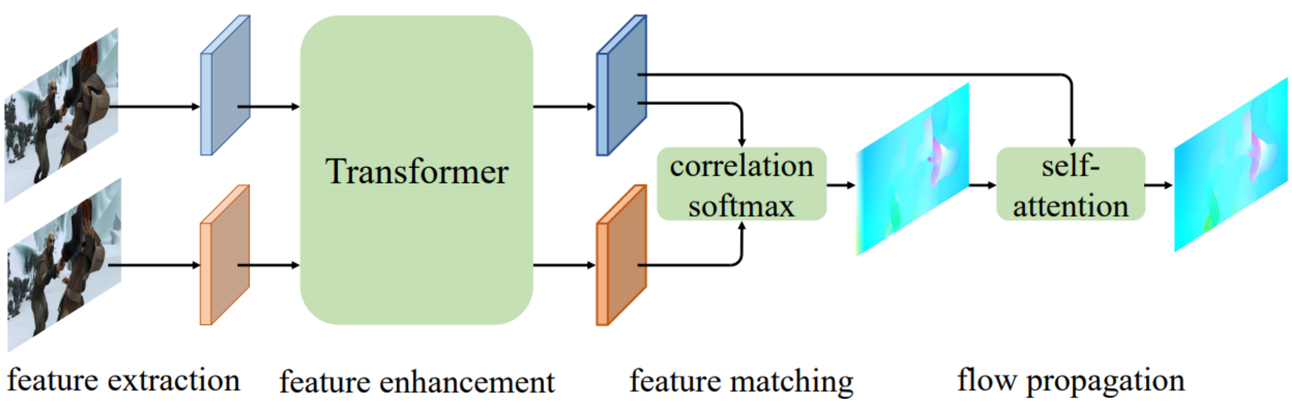
- □ 基于Transformer的方法：GMFlow
  - ■ Global correlation matching



(a) previous flow estimation approach



(b) GMFlow



[1] Xu, et al. “GMFlow: Learning Optical Flow via Global Matching.” In CVPR 2022. **(Oral)**

基于图像对的光流估计

- 基于Transformer的方法：GMFlow
  - Best performance and efficiency than RAFT

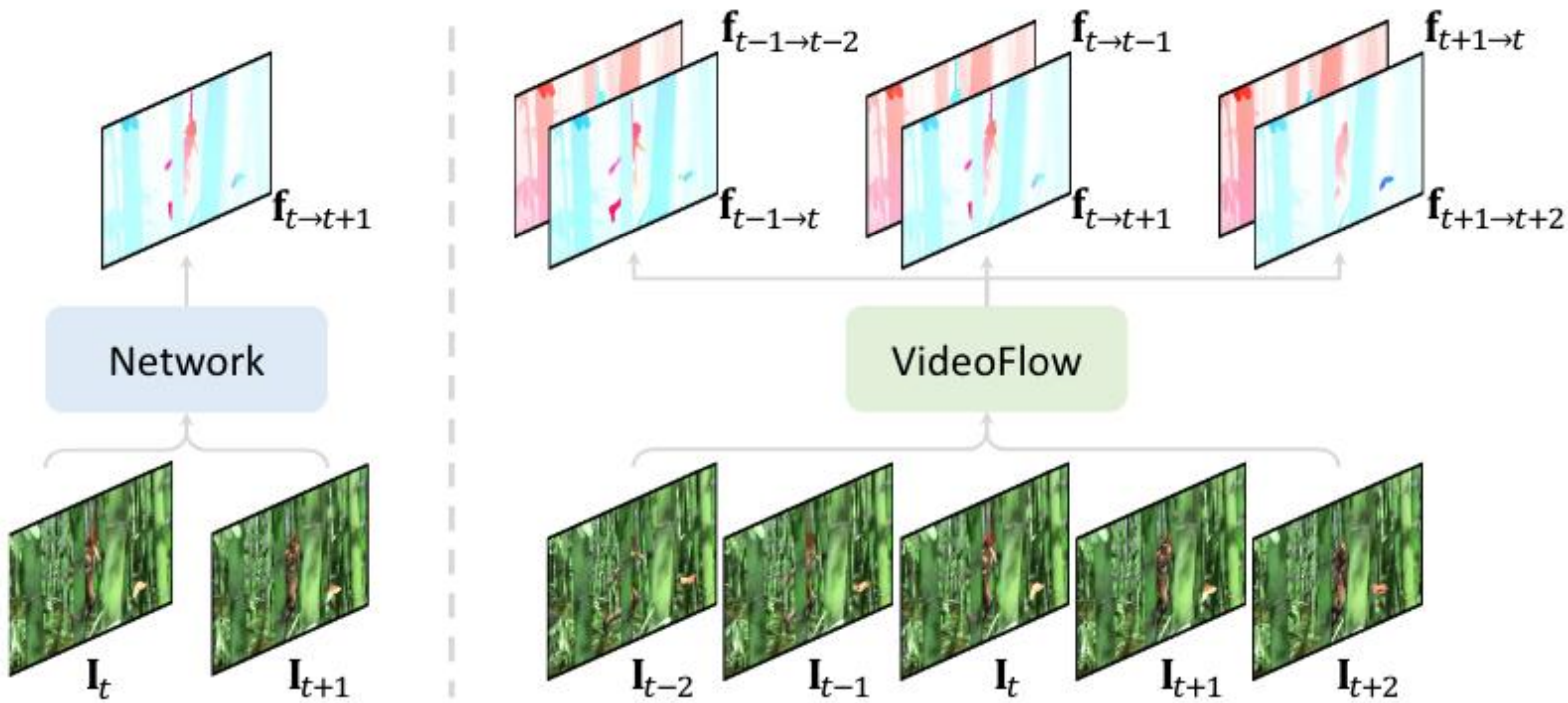
Method	#refine.	Things (val, clean)				Sintel (train, clean)				Sintel (train, final)				Param (M)	Time (ms)
		EPE	s_{0-10}	s_{10-40}	s_{40+}	EPE	s_{0-10}	s_{10-40}	s_{40+}	EPE	s_{0-10}	s_{10-40}	s_{40+}		
RAFT [39]	0	14.28	1.47	3.62	40.48	4.04	0.77	4.30	26.66	5.45	0.99	6.30	35.19	5.3	25 (14)
	3	6.27	0.69	1.67	17.63	1.92	0.47	2.32	11.37	3.25	0.65	4.00	20.04		39 (21)
	7	4.66	0.55	1.38	12.87	1.61	0.39	1.90	9.61	2.80	0.53	3.30	17.76		58 (31)
	11	4.31	0.53	1.33	11.79	1.55	0.41	1.73	9.19	2.72	0.52	3.12	17.43		78 (41)
	23	4.22	0.53	1.32	11.52	1.47	0.36	1.63	9.00	2.69	0.52	3.05	17.28		133 (71)
	31	4.25	**0.53**	1.31	11.63	1.41	0.32	1.55	8.83	2.69	0.52	3.00	17.45		170 (91)
GMFlow	0	3.48	0.67	1.31	8.97	1.50	0.46	1.77	8.26	2.96	0.72	3.45	17.70	4.7	57 (26)
	1	**2.80**	**0.53**	**1.01**	**7.31**	**1.08**	**0.30**	**1.25**	**6.26**	**2.48**	**0.51**	**2.81**	**15.67**	4.7	151 (66)

Table 3. **RAFT's iterative refinement framework *vs.* our GMFlow framework**. The models are trained on Chairs and Things training sets. We use RAFT's officially released model for evaluation. The inference time is measured on a single V100 and A100 (in parentheses) GPU at Sintel resolution (436×1024). Our framework gains more speedup than RAFT ($2.29\times$ *vs.* $1.87\times$, *i.e.*, ours: $151 \rightarrow 66$, RAFT: $170 \rightarrow 91$) on the high-end A100 GPU since our method doesn't require a large number of sequential computation.

[1] Xu, et al. "GMFlow: Learning Optical Flow via Global Matching." In CVPR 2022. **(Oral)**

基于多帧的光流估计

- VideoFlow
  - 同时对多帧估计双向光流



[1] Shi, et al. "VideoFlow: Exploiting Temporal Cues for Multi-frame Optical Flow Estimation." In ICCV 2023. **(Oral)**

基于多帧的光流估计

- VideoFlow
  - Using three frames as an example
  - Other structures similar to RAFT

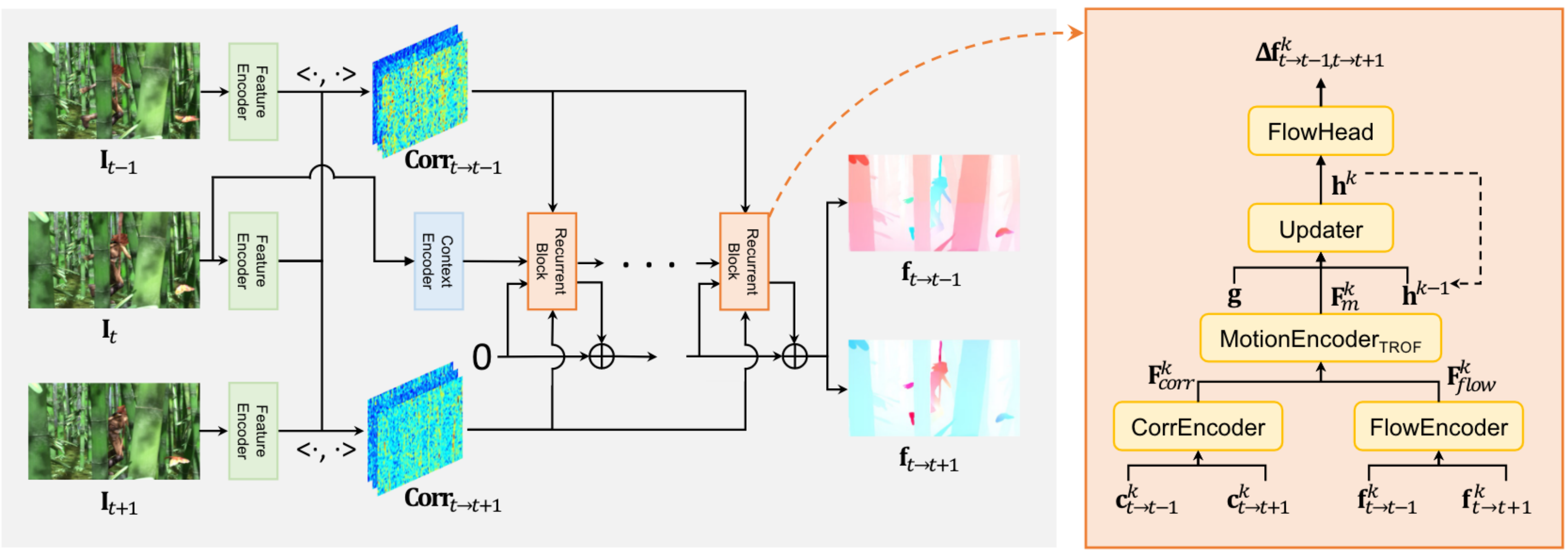
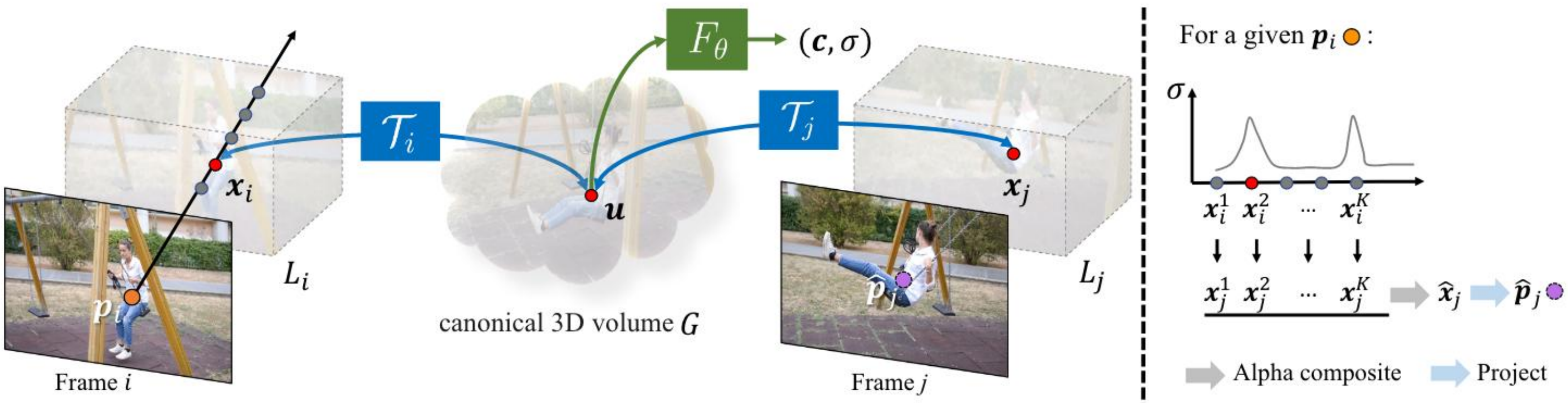


Figure 2. **Overview of VideoFlow in the three-frame setting.** Given a triplet of frames as input, VideoFlow jointly estimates bi-directional optical flows from the center frame to the adjacent previous and next frames. After building dual cost volumes, it recurrently fuses bi-directional flow features and correlation features to update flow predictions. The orange block on the right illustrates the recurrent flow refinement block.

[1] Shi, et al. "VideoFlow: Exploiting Temporal Cues for Multi-frame Optical Flow Estimation." In ICCV 2023. **(Oral)**

基于三维表征的光流估计

- OmniMotion
  - 从2D回到3D
  - 3D空间不存在遮挡问题



(a) OmniMotion representation (b) Computing 2D motion

[1] Wang, et al. "Tracking Everything Everywhere All at Once." In ICCV 2023. **(Best student paper)**

基于三维表征的光流估计

- OmniMotion
  - 从表征中提取的伪深度图，说明确实可以把深度估计得较为准确

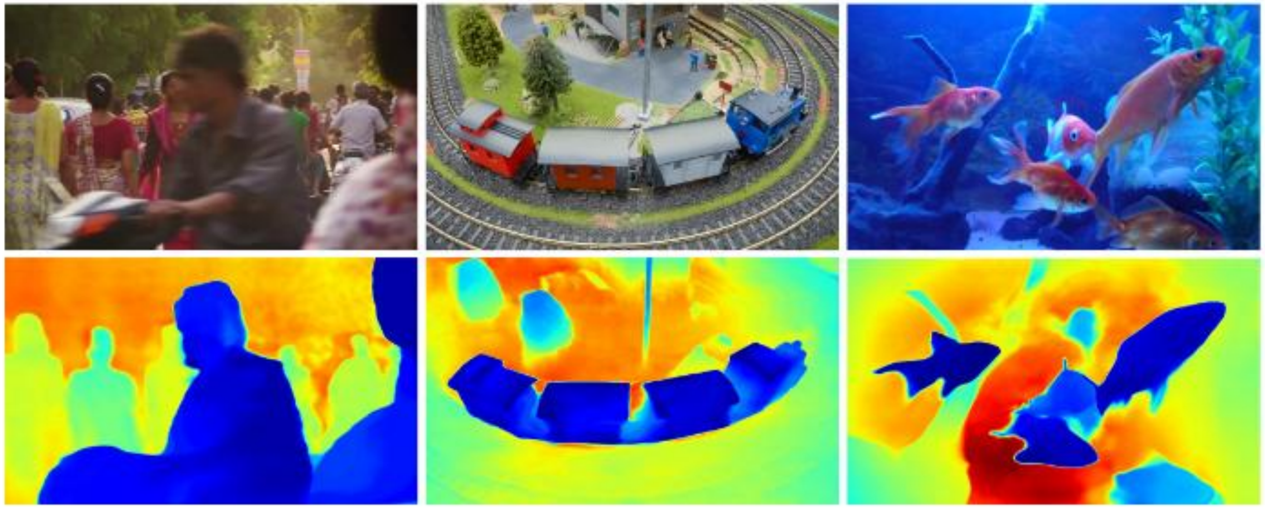


Figure 4: Pseudo-depth maps extracted from our representation, where blue indicates closer objects and red indicates further.

[1] Wang, et al. "Tracking Everything Everywhere All at Once." In ICCV 2023. **(Best student paper)**