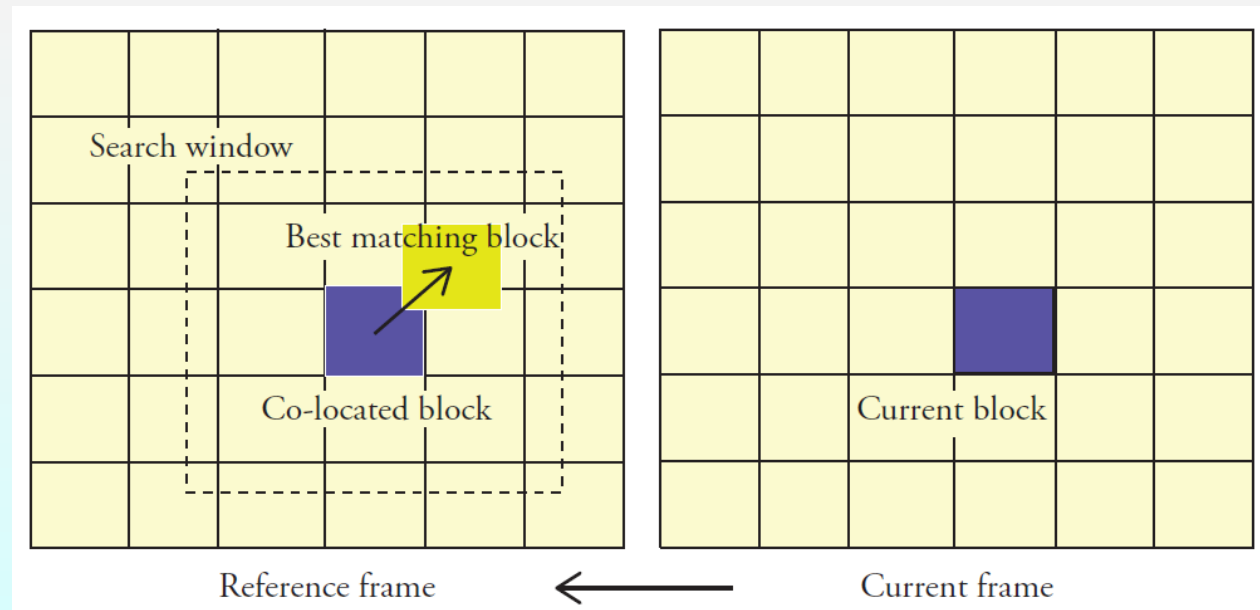


第12讲: 2D 运动估计中的匹配方法

- 块匹配
 - 基本块匹配
 - 快速搜索方法
 - 可变大小块匹配
 - 分层块匹配
- 相位相关
- 参数运动模型的估计
 - 线性运动模型的估计
 - 广义块匹配
- 单应性估计
 - DLT方法和坐标归一化

基本块匹配

- 块尺寸
- 搜索窗口尺寸
- 匹配准则包括：最小化均方误差 (MSE), 最小化平均绝对误差 (MAD), 最大化互相关, 最大化匹配像素数 (MPC), 等等。



基本块匹配 (cont'd)

- **MSE 准则** 定义为:

$$\text{MSE}(d_1, d_2) = \frac{1}{N_1 N_2} \sum_{(n_1, n_2) \in B} |s[n_1, n_2, k] - s[n_1 + d_1, n_2 + d_2, k + 1]|^2$$

其中 B 为 $N_1 \times N_2$ 块, (d_1, d_2) 为某个候选运动矢量 (MV)。

- **MAD 准则** 定义为:

$$\text{MAD}(d_1, d_2) = \frac{1}{N_1 N_2} \sum_{(n_1, n_2) \in B} |s[n_1, n_2, k] - s[n_1 + d_1, n_2 + d_2, k + 1]|$$

- **最终估计的 MV** 是对下式最小化时的 (d_1, d_2) :

$$\begin{bmatrix} \hat{d}_1 & \hat{d}_2 \end{bmatrix}^T = \arg \min_{(d_1, d_2)} \text{MSE}(d_1, d_2)$$

或

$$\begin{bmatrix} \hat{d}_1 & \hat{d}_2 \end{bmatrix}^T = \arg \min_{(d_1, d_2)} \text{MAD}(d_1, d_2)$$

或者简单使用: $\text{SAD}(d_1, d_2) = \sum_{(n_1, n_2) \in B} |s[n_1, n_2, k] - s[n_1 + d_1, n_2 + d_2, k + 1]|$

快速搜索方法

- **初始搜索像素预测**：可从空间或时间相邻块预测。
- **缩小搜索空间**：是多步搜索过程，定义了初始步长和以搜索窗口中心为中心的一组搜索点，并在步长减小到1时终止。
 - Jain和Jain提出的**对数搜索** [Jai 81]
 - 由Koga等人提出的**三步搜索** (TSS) [Kog 81]
 - **菱形搜索**
- **提前终止**
 - **连续消除算法 (SEA)** [Li 95]，基于**三角不等式**
- 通常使用上述方法的组合。

连续消除算法 (SEA)

- **SEA**是基于三角不等式[Li 95]提出的:

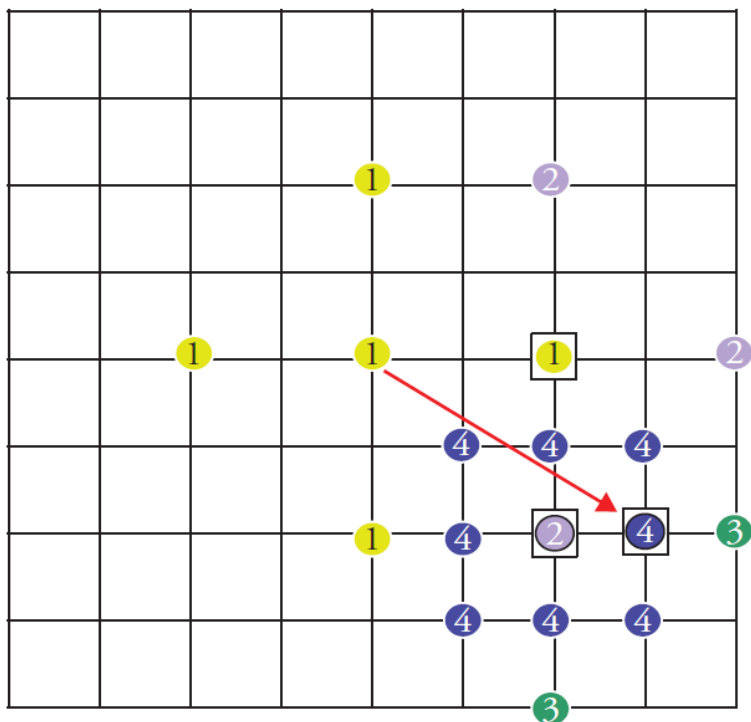
$$\begin{aligned} \sum_{(n_1, n_2) \in B} |S[n_1, n_2, k]| - \sum_{(n_1, n_2) \in B} |s[n_1 + d_1, n_2 + d_2, k + 1]| \\ \leq \sum_{(n_1, n_2) \in B} |s[n_1, n_2, k] - s[n_1 + d_1, n_2 + d_2, k + 1]| = \text{SAD}(d_1, d_2) \end{aligned}$$

其中 $\text{SAD}(d_1, d_2)$ 表示绝对误差和。

- 可以使用**box滤波技术**更快地计算强度值之和与位移强度值之和之间的差值，该差值作为**SAD值的一个下限**。
- 对每个候选MV，将其**SEA公式左侧的差值**与先前的最小**SAD值**进行比较，如果下界大于先前的最小值，则跳过SAD计算。

对数搜索

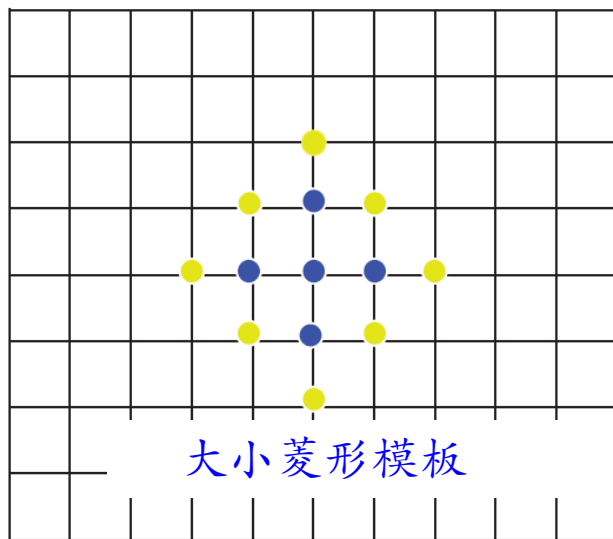
- **对数搜索** 开始于在搜索窗口的中心（零运动矢量）以及从水平和垂直方向 $\pm P$ 个像素的四个点（其中通常 $P = M/2$ ）计算SAD。



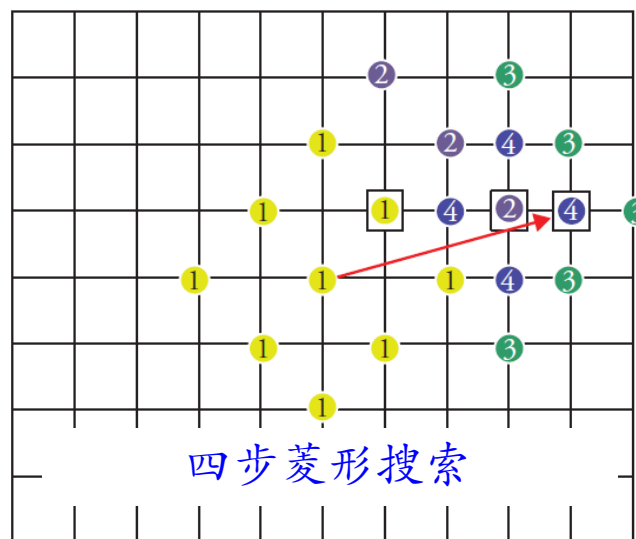
- 如果**最小SAD出现在中心**，则**步长减半**。否则，步长保持不变，下一搜索**以具有最小SAD的像素为中心**，计算新的四点 \pm 新像素 P 个像素的SAD值。
- 当**步长变为1**时，搜索关于当前中心的**8个相邻像素**，具有最小SAD的像素位置作为最终整数MV。

菱形搜索

- **钻石搜索 (DS)** [Zhu 00]采用**两个菱形**搜索模式：具有**9个搜索点**的大菱形和具有**5个点**的小菱形。
- DS从**以搜索窗口中心为中心的大菱形**开始。如果最小SAD不在中心发生，则下一搜索以具有最小SAD值的像素为中心，并且以大菱形继续搜索，直到**最小值出现在大菱形的中心**。



(a)



(b)

- 当**最小值出现在中心**时，使用以该中心像素为中心的小**菱形**进行最终搜索，而小菱形上的最小SAD的位置就是最终的整数MV。

亚像素运动估计

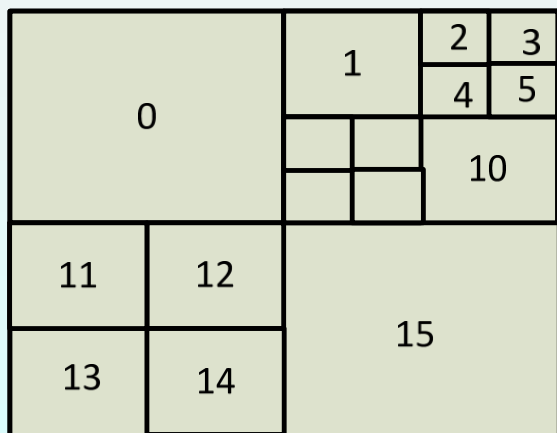
- 由于需要计算**像素之间的插值**，并需要计算更多数量的候选块，所以亚像素运动估计的**计算复杂度较高**。
 - 确定**最佳整数MV**
 - 确定关于最佳整数MV的**最佳半像素位置**
 - 确定关于最佳半像素MV的**最佳四分之一像素位置**。



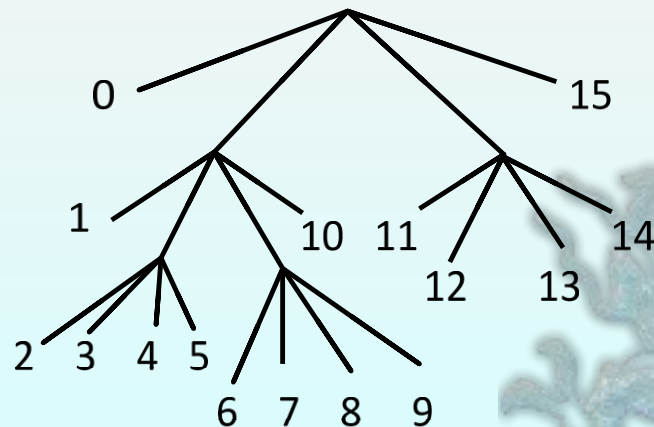
格点表示图像像素。

可变大小块匹配

- 较小的块用于具有复杂运动的图像区域，而较大的块可用于图像静止或经历均匀运动的区域。
- 高效实现VSBM的核心思想是尽可能复用SAD的计算。
 - 计算最小子块大小（例如， 4×4 ）的SAD，然后通过对这些SAD值的组合来计算较大块的SAD
 - 复用候选MV的部分SAD值以用于计算其他候选MV的SAD。



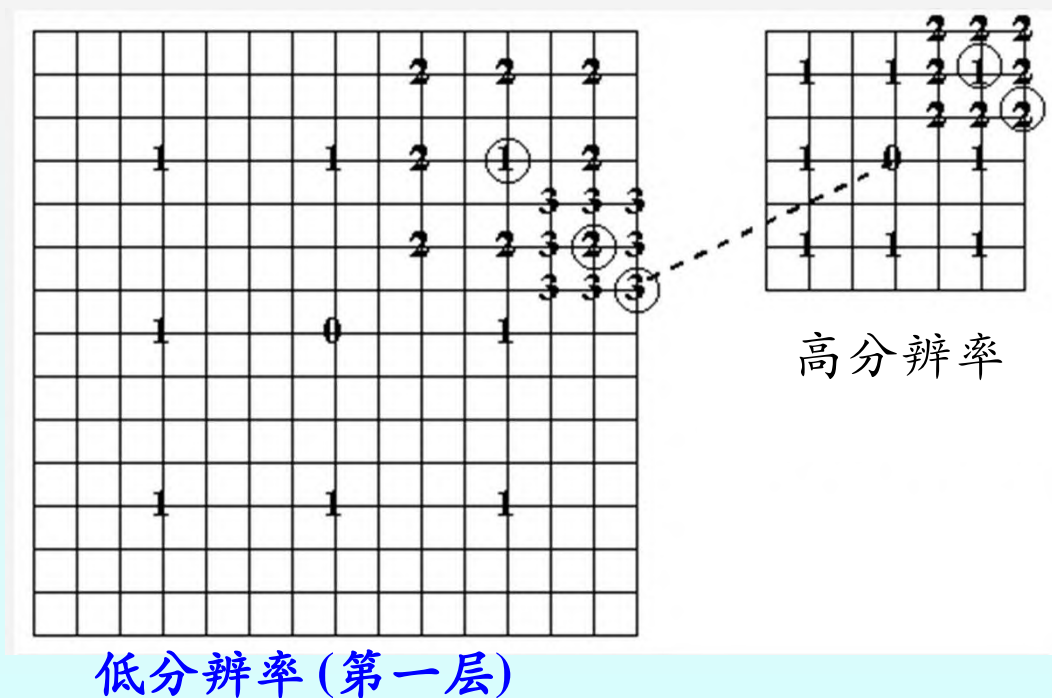
块的可变大小划分



划分的四叉树表示

分层块匹配

- 将较低分辨率级别的**MV估计传递**到下一较高分辨率级别作为初始估计。
- 较高的分辨率水平是用来以**相对较小的窗口尺寸**和**较好的初始估计**来**精细调整MV**估计。



- 如果在较低分辨率级别下**多于一个MV产生相似的SAD**，则它们可以在较高级别被传递和细化，并最终从其中选择最佳MV。

相位相关法

- 首先由[Kug 75]提出的**相位相关**方法利用了以下事实：**图像域中的平移**对应于**2D空间频率域中的线性相移**，因为 $S_{k+1}(\mathbf{x}) = S_k(\mathbf{x} + \mathbf{d}(\mathbf{x}))$ 隐含：

$$S_{k+1}(f_1, f_2) = e^{-j(d_1 f_1 + d_2 f_2)} S_k(f_1, f_2)$$

- 两帧之间的傅里叶相位差**的线性项可确定运动估计。**相位相关函数**由下式给出：

$$C_{k,k+1}(f_1, f_2) = \frac{S_{k+1}(f_1, f_2) S_k^*(f_1, f_2)}{|S_{k+1}(f_1, f_2) S_k^*(f_1, f_2)|}$$

其中 $S_k(f_1, f_2)$ 是帧k相对于空间变量 x_1 和 x_2 的2D傅立叶变换， $*$ 表示复共轭。

- 该方法对**帧到帧的强度偏移**（偏置或乘以常数）不敏感，因为它们不影响傅里叶相位。

相位相关法 (cont'd)

- 如果在帧k和k + 1之间存在**平移运动**，那么：

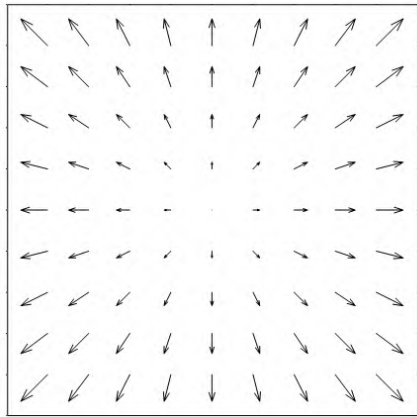
$$C_{k,k+1}(f_1, f_2) = e^{-j(d_1 f_1 + d_2 f_2)}$$

其2-D傅里叶反变换： $c_{k,k+1}[n_1, n_2] = \delta[n_1 - d_1, n_2 - d_2]$

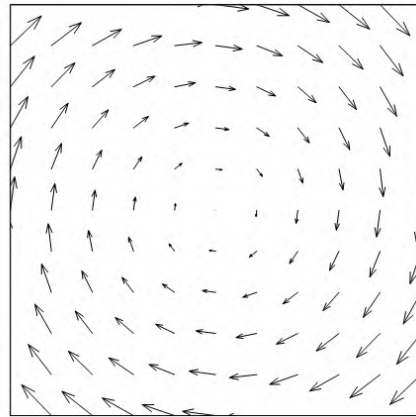
脉冲位置指示了**位移矢量** (d_1, d_2)。

- 最大位移估计/块尺寸**：由于DFT是周期性的块尺寸 $N_1 \times N_2$ ，并且实值图像的DFT呈现Hermitian对称性，位移估计的最大范围被限制为 $[-(N_i/2) + 1, N_i/2]$ ， N_i 为偶数。例如，为了估计在范围 $[-31, 32]$ 内的位移，块大小应该至少为 64×64 。
- 边界效应**：为了在逆2D DFT中获得完美的脉冲，对于每个块，移位必须是循环的。由于在块的一端消失的部分通常不会在另一端重现，所以脉冲将退化为由于具有矩形核的窗口而导致的峰值。
- 多个运动**：多个峰值。

参数运动模型

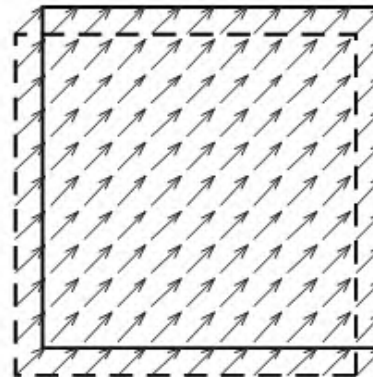


摄像机变焦

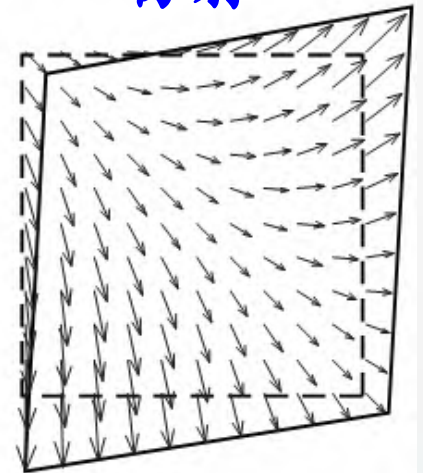


摄像机绕Z轴旋转

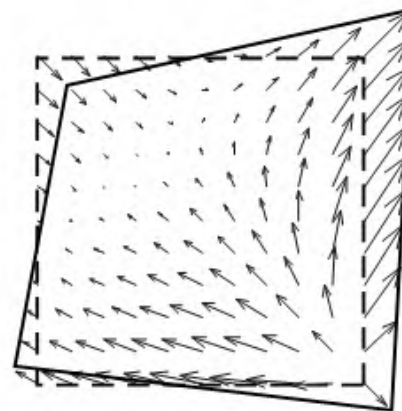
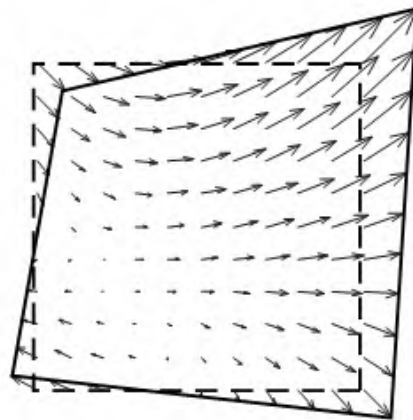
平移



仿射



双线性



透视投影

参数运动模型的估计

- 根据像素对应性的仿射参数估计：对于点对应 $[x_1 \ x_2] \rightarrow [x'_1 \ x'_2]$ ，**仿射映射**可以表示为：

$$\begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & x_2 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix}$$

- 给定 **$N \geq 3$ 对点对应**，可得到由 **$2N$ 个线性方程**组成的方程组，

$$\mathbf{y}_{2N} = \mathbf{H}_{2N \times 6} \mathbf{a}_6$$

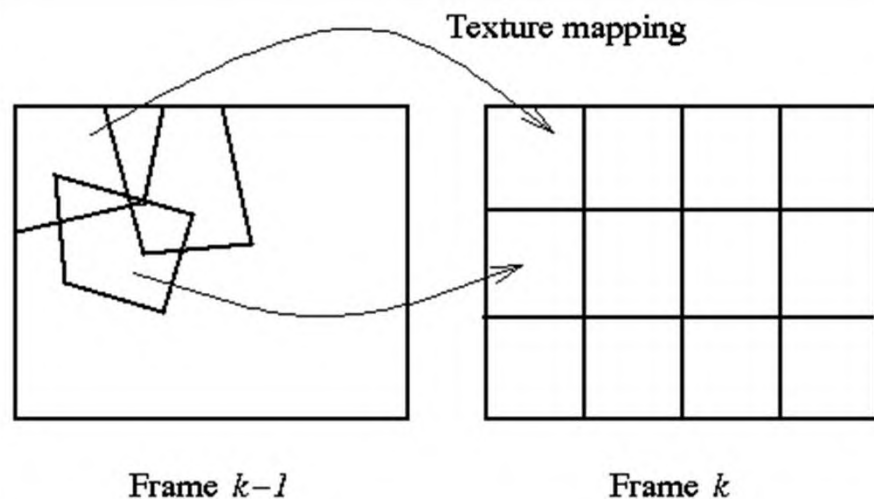
共有 6 个未知参数，可使用**最小二乘法**求解。

- 最小二乘法的**解**为：

$$\mathbf{a}_6 = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y}$$

广义块匹配

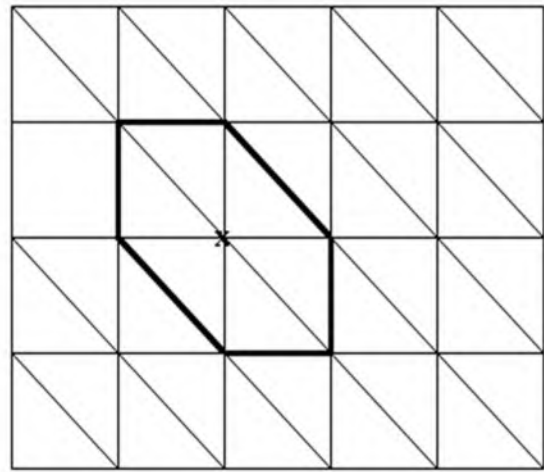
- 无连接约束的块匹配方法可以概括为：
 - 将当前帧划分为**矩形块**。
 - 将参考帧中同样位置的块的四个角点的坐标从初始猜测开始更新，以形成**候选匹配的四边形**。
 - 对于每个候选者，使用在四个角点的对应，找到将该四边形映射到当前帧中的**矩形块上的模型参数**。



- 使用**计算的参数模型执行空间变换**，并计算给定块和匹配的四边形之间的MSE。
- 选择产生**最小MSE或MAD**的空间变换。

基于网格运动估计

- **六边形搜索** 是基于以下观察：在均匀三角形网格中的每个节点处有六条线相交，并且这六个三角形的边界将确定一个六边形。
- 为了保持**连通性**，我们假设每个节点处的运动矢量被约束为**位于该六边形内**。然后，每个节点（在参考帧中）**每次只有一个被扰动**，以找到当前帧和参考帧之间的最佳匹配六边形。



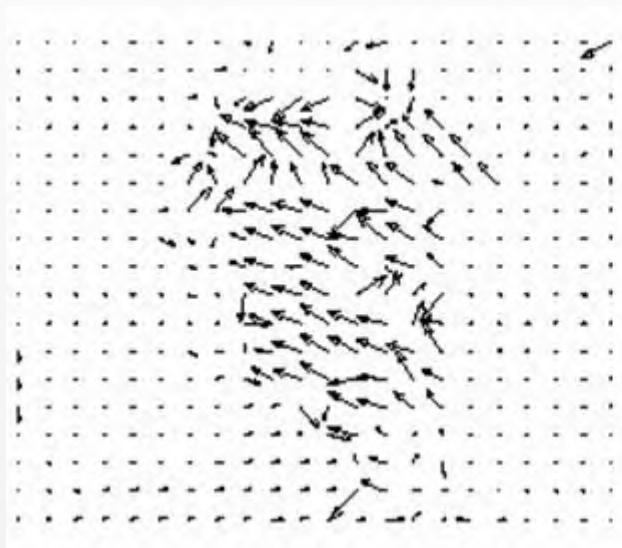
- 通过使用它们**各自的仿射参数**由受扰动节点影响的六个三角形中的每个三角形内的仿射纹理映射来计算六边形的SAD。

基于网格运动估计

参考帧



当前帧



全搜索块匹配 (半像素精度)



预测图像



基于网格

单应估计

- 对于每个预定特征点 i ，我们在齐次坐标 \mathbf{z} 中表示投影变换，其中

$$\mathbf{z}_i = [z_{i1} \ z_{i2} \ z_{i3}]^T$$

$$x_{i1} = z_{i1}/z_{i3}, \ x_{i2} = z_{i2}/z_{i3}$$

$$\mathbf{z}'_i = \begin{bmatrix} z'_{i1} \\ z'_{i2} \\ z'_{i3} \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x_{i1} \\ x_{i2} \\ 1 \end{bmatrix} = \mathbf{H} \mathbf{z}_i, \quad i = 1, \dots, N$$

而 z_{i3} 设置为1。

- 观察到 \mathbf{H} 在乘以非零常数时不会改变图像坐标 $x'_{i1} = z'_{i1}/z'_{i3}$ ， $x'_{i2} = z'_{i2}/z'_{i3}$ ，这被称为**尺度模糊度**。因此， \mathbf{H} 是具有仅**8个自由度的均匀矩阵**，即使其共具有9项。

- 图像坐标：**

$$x'_{i1} = \frac{z'_{i1}}{z'_{i3}} = \frac{h_1 x_{i1} + h_2 x_{i2} + h_3}{h_7 x_{i1} + h_8 x_{i2} + h_9}$$

$$x'_{i2} = \frac{z'_{i2}}{z'_{i3}} = \frac{h_4 x_{i1} + h_5 x_{i2} + h_6}{h_7 x_{i1} + h_8 x_{i2} + h_9}$$

单应估计 (cont'd)

- 消去分母，整理可得：

$$h_7 x'_{i1} x_{i1} + h_8 x'_{i1} x_{i2} + h_9 x'_{i1} - h_1 x_{i1} - h_2 x_{i2} - h_3 = 0$$

$$h_7 x'_{i2} x_{i1} + h_8 x'_{i2} x_{i2} + h_9 x'_{i2} - h_4 x_{i1} - h_5 x_{i2} - h_6 = 0$$

也可以写成**向量-矩阵形式**：

$$\mathbf{A}_i \mathbf{h} = \begin{bmatrix} -x_{i1} & -x_{i2} & -1 & 0 & 0 & 0 & x'_{i2} x_{i1} & x'_{i2} x_{i2} & x'_{i1} \\ 0 & 0 & 0 & -x_{i1} & -x_{i2} & -1 & x'_{i2} x_{i1} & x'_{i2} x_{i2} & x'_{i2} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \mathbf{0}$$

- 因此，可获得用于特征点对应的两个线性独立的方程，条件是**排除无穷远点**，即假设 $z'_{i3} \neq 0$ 。

单应估计 (cont'd)

- 直接线性变换 (DLT) 方法: 给定 $N \geq 4$ 个特征点对应, 使得没有三个点是共线的, 我们可以获得 $2N$ 个齐次线性方程:

$$\begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_N \end{bmatrix} \mathbf{h} = \mathbf{A} \mathbf{h} = \mathbf{0}$$

其中 \mathbf{A} 是 $2N \times 9$ 矩阵, \mathbf{h} 是由未知单应性参数组成的 9×1 向量。

- 如果 $N = 4$ 或 $N > 4$, 但所有点对应都是**精确的 (无噪声)**, 则 \mathbf{A} 的秩是8。因此, \mathbf{A} 具有一维零空间, 可以给出 \mathbf{h} 的解, 该解最多有一个非零尺度参数。回顾可知, \mathbf{H} 被定义为一个比例因子。可以通过设置 $\|\mathbf{h}\| = 1$ 来任意地选择比例因子, 以避免平凡解 $\mathbf{h} = \mathbf{0}$ 。如果 $N > 4$ 并且特征点对应是有噪的, 则超定系统 $\mathbf{A} \mathbf{h} = \mathbf{0}$ 是不一致的, 且无解。

单应估计 (cont'd)

- 此时可以找到 **\mathbf{h} 的最小二乘解**，使在 $\|\mathbf{h}\|=1$ 的条件下最小化 $\|\mathbf{A}\mathbf{h}\|$ 。在任一情况下， \mathbf{h} 由 \mathbf{V} 的最后一列给出。
其中 $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ 为 \mathbf{A} 的**奇异值分解(SVD)**。
- 当**不可能去均质**时，另外一种 DLT 方法的推导是**基于 $\mathbf{z}_i' \times \mathbf{H} \mathbf{z}_i = 0$ 这个事实**且使所有三个方程都包括 $\mathbf{z}_{3i}' = 0$ (无穷远点)的情况 [Har 04]。
- 请注意，可以通过首先设置 **$\mathbf{h}_9 = 1$ 作为尺度参数**并求解所得到的非齐次方程组来估计单应性的参数。然而，如果 \mathbf{h}_9 的实际值接近0，则该方法的结果将很差，因此并不推荐。

单应估计 (cont'd)

- 归一化: 基本DLT算法的性能取决于两帧图像坐标系的**原点**和**尺度**。因此, 像素坐标的归一化有助于获得**稳定的数值解** [Har 04]。归一化DLT算法步骤如下:
 - 计算第一帧 $\tilde{\mathbf{x}}_i = \mathbf{T}\mathbf{x}_i$ 相似变换, 使得新坐标系的原点是点 \mathbf{x}_i 的质心, 并且点 \mathbf{x}_i 与原点的平均距离为 $\sqrt{2}$ 。
 - 不依赖第一帧, 用相同方法计算第二帧的**相似变换** $\tilde{\mathbf{x}}'_i = \mathbf{T}'\mathbf{x}'_i$ 。
 - 用DLT算法得到点对应 $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}'_i)$ 的单应性 $\tilde{\mathbf{H}}$ 。
 - 期望的单应矩阵 $\mathbf{H} = (\mathbf{T}')^{-1}\tilde{\mathbf{H}}\mathbf{T}$ 。
- **标准化DLT方法**, 也称为归一化8点算法, 可得到令人满意的结果 [Har 97]。