

第十四章 二维运动估计

早期设计的机器视觉系统主要是针对静态场景的, 为了满足更高级的应用需求, 必须研究用于动态场景分析的机器视觉系统. 动态场景分析视觉系统一般需要较大的存储空间和较快的计算速度, 因为系统的输入是反应场景动态变化的图像序列, 其包含的数据十分巨大. 图像动态变化可能由摄像机运动、物体运动或光照改变引起, 也可能由物体结构、大小或形状变化引起. 为了简化分析, 通常我们假设场景变化是由摄像机运动和物体运动引起的, 并假设物体是刚性的.

根据摄像机和场景是否运动将运动分析划分为四种模式: 摄像机静止—物体静止, 摄像机静止—物体运动, 摄像机运动—物体静止, 摄像机运动—物体运动, 每一种模式需要不同的分析方法和算法. 摄像机静止—物体静止模式属于简单的静态场景分析. 摄像机静止—场景运动是一类非常重要的动态场景分析, 包括运动目标检测、目标运动特性估计等, 主要用于预警、监视、目标跟踪等场合. 摄像机运动—物体静止是另一类非常重要的动态场景分析, 包括基于运动的场景分析、理解, 三维运动分析等, 主要用于移动机器人视觉导航、目标自动锁定与识别等. 在动态场景分析中, 摄像机运动—物体运动是最一般的情况, 也是最难的问题, 目前对该问题研究的还很少.

图像运动估计是动态场景分析的基础, 现在已经成为计算机视觉新的研究热点. 根据所涉及的空间, 将图像运动估计分为二维运动估计和三维运动估计, 显然, 这种划分不是十分严格, 因为二维运动参数的求解有时需要三维空间的有关参数引导, 而许多三维参数的求解需要以二维参数为基础. 本章主要讨论二维运动估计, 三维运动估计和分析将在第十五章讨论.

14.1 图像运动特征检测

对许多应用来说, 检测图像序列中相邻两帧图像的差异是非常重要的步骤. 场景中任何可察觉的运动都会体现在场景图像序列的变化上, 如能检测这种变化, 就可以分析其运动特性. 如果物体的运动限制在平行于图像平面的一个平面上, 则可以得到物体运动特性定量参数的很好估计. 对于三维运动, 则只能得到物体空间运动的定性参数估计. 场景中光照的变化也会引起图像强度值的变化, 有时会引起较大的变化. 动态场景分析的许多技术都是基于对图像序列变化的检测. 检测图像变化可以在不同的层次上进行, 如像素、边缘或区域. 在像素层次上要对所有可能的变化进行检测, 以便在后处理阶段或更高层次上使用.

14.1.1 差分图像

检测图像序列相邻两帧之间变化的最简单方法是直接比较两帧图像对应像素点的灰度值. 在这种最简单的形式下, 帧 $f(x, y, j)$ 与帧 $f(x, y, k)$ 之间的变化可用一个二值差分图像 $DP_{jk}f(x, y)$ 表示:

$$DP_{jk}f(x, y) = \begin{cases} 1 & \text{如果 } |f(x, y, j) - f(x, y, k)| > T \\ 0 & \text{其它} \end{cases} \quad (14.1)$$

式中 T 是阈值.

在差分图像中, 取值为 1 的像素点被认为是物体运动或光照变化的结果. 这里假设帧与帧之间配准或套准得很好. 图 14.1 和 14.2 示意了两种图像变化情况, 一种是由于光照变化造成的图像变化, 另一种是由于物体的运动产生的图像变化. 需要指出, 阈值在

这里起着非常重要的作用。对于缓慢运动的物体和缓慢光强变化引起的图像变化，在一个给定的阈值下可能检测不到。

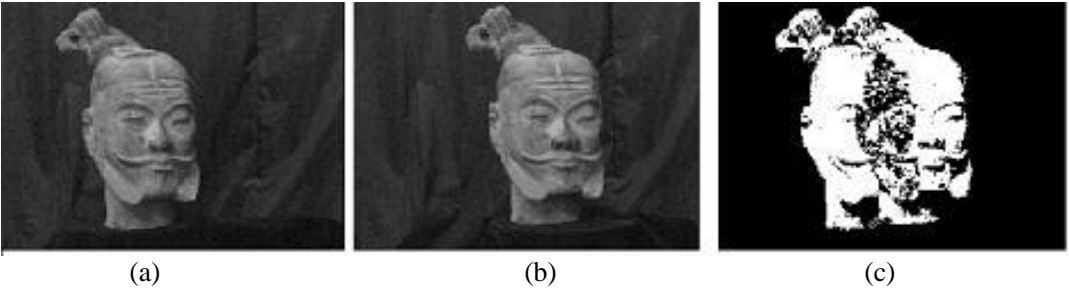


图 14.1 (a)和(b)是取自一个运动摄像机获取的静态场景图像序列的两帧图像，
(c)是它们的差分图像($T=40$)



图 14.2 (a)和(b)是取自光照变化的图像序列的两帧图像，
(c)是它们的差分图像($T=80$)

(1) 尺度滤波器

在实际中，使用上述差分方法计算的差分图像经常会含有许多噪声。一个简单噪声消除方法是使用尺度滤波器，滤掉小于某一尺度的连通成分，因为这些像素常常是由噪声产生的，留下大于某一尺度阈值的 4-连通或 8-连通成分，以便作进一步的分析。对于运动检测，这个滤波器非常有效。但也会将一些有用的信号滤掉，比如那些来自于缓慢运动或微小运动物体的信号。在图 14.3 中，我们给出了图 14.1、14.2 图像差分图像的尺度滤波结果。



图 14.3 (a)图 14.1 差分图像的尺度滤波结果
(b)图 14.2 差分图像的尺度滤波结果

(2) 鲁棒检测方法

为了使图像变化检测更鲁棒，可以使用统计方法或基于强度分布的局部逼近方法来比较两帧图像之间的光强特性，比如使用第三章讨论过的似然比

$$\lambda = \frac{\left[\frac{\sigma_1 + \sigma_2}{2} + \left(\frac{\mu_1 - \mu_2}{2} \right)^2 \right]^2}{\sigma_1 * \sigma_2} \quad (14.2)$$

来进行两帧图像之间的比较, (式中 μ 和 σ 表示区域的平均灰度和方差), 然后计算差分图像:

$$DP_{jk}f(x, y) = \begin{cases} 1 & \text{如果 } \lambda > T \\ 0 & \text{其它} \end{cases} \quad (14.3)$$

式中 T 是阈值. 对于许多真实场景, 将似然比和尺度滤波组合起来使用是非常有效的.

上面讨论的似然比测试是基于区域服从均匀二阶统计的假设. 如果使用小平面和二次平面来近似这些区域, 似然比测试方法的性能还能有显著的提高. 注意, 似然比测试是在像素层次上检测相似度, 因此只能确定所考察的区域是否有同样的灰度特征; 有关区域的相对强度信息则没有得到保留. 使用似然比方法检测图像序列中的运动部分可能增加计算量.

(3) 累积差分图像

缓慢运动物体在图像中的变化量是一个很小的量, 尺度滤波器可能会将这些微小量当作噪声滤掉. 当使用鲁棒检测方法时, 因为是基于区域的变化检测, 因此会使得检测小位移或缓慢运动物体的问题变得更加严重.

解决这一问题的一种方法是累积差分图像方法(accumulative difference picture, ADP), 其基本思想是通过分析整个图像序列的变化(而不是仅仅分析两帧之间的变化)来检测小位移或缓慢运动物体. 这种方法不仅能用来可靠检测微小运动或缓慢运动的物体, 也可用来估计物体移动速度的大小和方向以及物体尺度的大小. 累积差分图像可分为一阶累积差分图像(FADP)和二阶累积差分图像(SADP).

一阶累积差分图像的形成过程如下: 将图像序列的每一帧图像与一幅参考图像进行比较, 当差值大于某一阈值时, 就在累积差分图像中加 1. 通常将图像序列的第一帧作为参考图像, 并且置累积差分图像 $FADP_0$ 的初始值为 0. 这样, 在第 k 帧图像上的累积差分图像 $FADP_k(x, y)$ 为:

$$FADP_k(x, y) = \begin{cases} 0 & k = 0 \\ FADP_{k-1}(x, y) + DP_{1k}(x, y) & \text{其它} \end{cases} \quad (14.4)$$

图 14.4 是利用累积差分图像检测的结果示意图.

二阶差分图像的构造为: 对应于第 n 帧 ($n = 0, 1, \dots, N$ 且 $n > 1$) 的二阶差分图像在 (x, y) 位置的值为 “1”, 表明在这个位置上第 $n-1$ 帧和第 n 帧的一阶差分图像 $FADP$ 具有不同的符号.

$$SADP_k(x, y) = \begin{cases} 1 & SADP_{k-1}(x, y) \neq SADP_k(x, y) \quad k > 1 \\ 0 & \text{其它} \end{cases} \quad (14.5)$$

累积差分图像具有许多性质, 可以用于描述物体运动的总体参数[Jain]

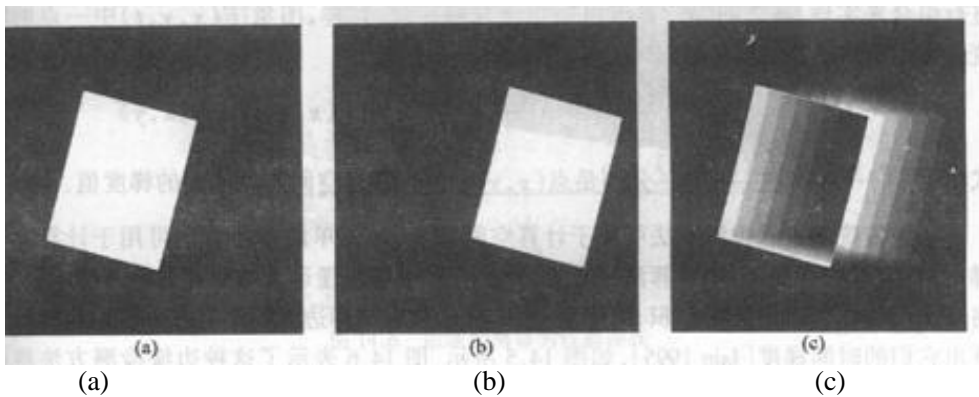


图 14.4 (a) 第一帧图像 (参考图像), (b) 第三帧后的累积差分图像, (c) 最后一帧后的累积差分图像

用于运动检测的差分图像的最大特点是它的简单性, 但差分图像极易受噪声污染. 照明和摄像机位置的改变, 以及摄像机的电子噪声, 都会产生很多错误数据. 将似然比和尺度滤波器组合起来能消除大部分摄像机噪声. 照明的改变会给所有基于光强的运动检测方法带来问题, 这些问题有可能在后期的分析和理解层次上得以解决. 图像序列各帧之间的错误套准也会导致错误数据, 如果套准错误不十分严重, 则累积差分图像还是可以消除这些错误数据的.

必须强调, 在像素层次上检测不相似度只能通过检测光强变化来实现. 在动态场景分析中, 这是最低层次的分析. 在检测完像素变化以后, 还需要通过其它的处理过程来解释这些变化. 经验表明, 差分图像最有效的应用是对图像进行概略处理, 以便将解释的注意力引向场景中出现“活动”的区域. 场景中事件的粗略信息也可由差分图像中的某些特征来提取.

14.1.2 时变边缘检测

我们知道, 边缘在静态场景图像分析中起着十分重要的作用, 因此有理由推测时变边缘在动态场景图像分析中也是非常重要的. 在进行图像分割与匹配方法中, 人们将精力主要集中在静态特征与运动特征之间的匹配. 实际上, 这些静态特征是运动信息提取的最大障碍. 如果直接检测运动特征, 那么完成匹配所需的计算量可以根本上大大减小.

一条边缘运动后仍然是一条边缘. 运动边缘是通过逻辑“与”算子对时间和空间梯度进行组合来实现, 其中的“与”算子可以由乘法来完成. 这样, 图像 $E(x, y, t)$ 中一点的时变边缘由下式给出:

$$\begin{aligned} E_t(x, y, t) &= \frac{dE(x, y, t)}{dS} \cdot \frac{dE(x, y, t)}{dt} \\ &= E(x, y, t) \cdot D(x, y) \end{aligned} \quad (14.6)$$

式中 $\frac{dE(x, y, t)}{dS}$ 和 $\frac{dE(x, y, t)}{dt}$ 分别是点 (x, y, t) 的光强在空间和时间上的梯度值.

各种传统的边缘检测方法可用于计算空间梯度, 而简单的差分方法可用于计算时间梯度. 在大多数情况下, 边缘算法很有效. 为了克服遗漏缓慢运动边缘和弱边缘的问题, 可将一个阈值作用于上式的乘积, 而不是一阶差分, 然后使用边缘检测器或一阶检测边缘器算出它们的时间梯度[Jain], 如图 14.5 和 14.6 所示. 由图 14.6 可见, 这种边缘检测方法将对有清晰边缘的缓慢运动和以适当速度运动的弱边缘响应. 这种检测方法的另一个重要特点是不需要对任何位移大小作出假设. 当边缘的运动非常大时, 检测器的性能也是十分满意的.

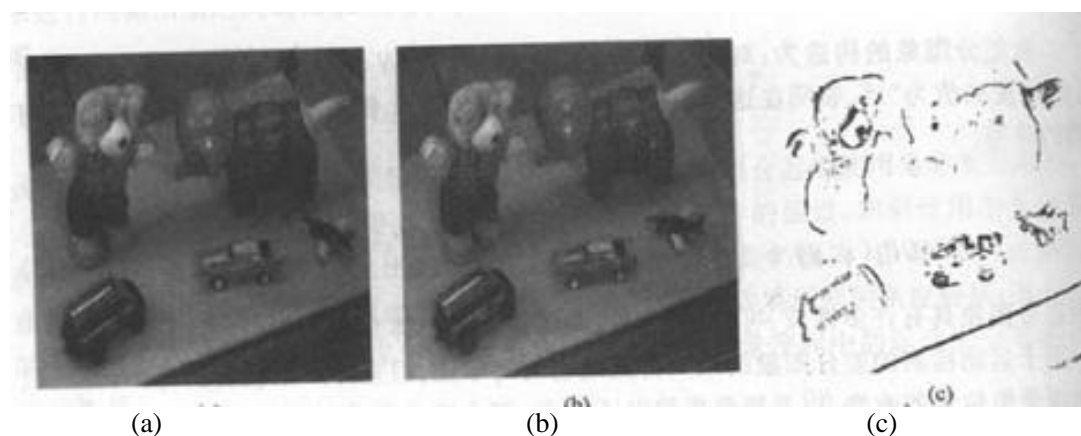


图 14.5 时变边缘检测器运行结果示意图

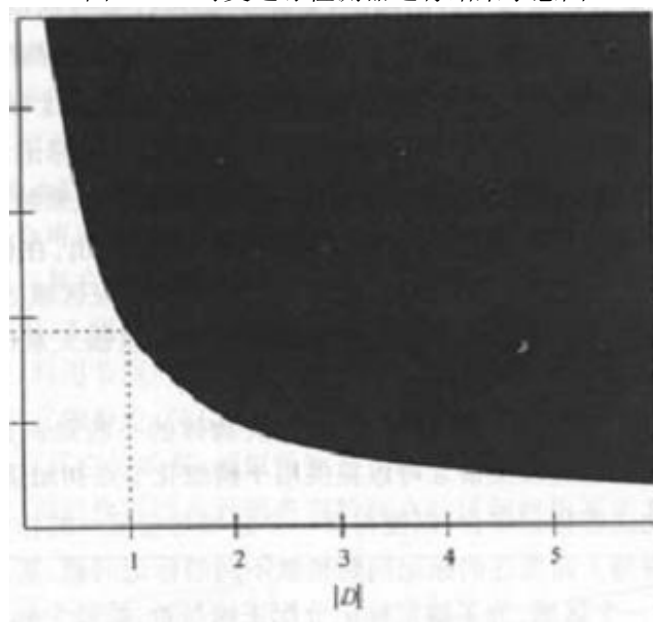


图 14.6 边缘检测器的性能曲线

14.1.3 运动对应性

已知一个图像序列，我们可以分析并确定序列中每一帧图像上的特征点。为估计图像运动特性，必须在图像各帧之间建立这些特征点的对应关系。运动图像的对应问题与立体视觉中的对应问题相似，不过立体视觉使用的约束主要是外极线约束，运动图像对应问题使用的是其它类型的约束。下面讨论一种约束传播方法来解决对应问题。

(1) 松弛标记

标记(labeling)是指将一组已知标记分配给场景中对应的各个物体。标记问题可以表示为如图 14.7 所示的形式。每一个节点表示一个标记区域(或物体)，连结节点的弧线表示区域间的关系。假定每一个节点上有一个处理器，在每一个节点上定义集合 R , C , L 和 P 。集合 R 包含节点间所有可能的关系(relation)；集合 C 表示这些关系的相容性(compatibility)，相容性将有助于对关系 R 进行约束以及对图像中每个区域的标记进行约束；集合 L 包含所有指定给该节点的标记；集合 P 表示计算过程中赋予节点的所有可能的层次。假定在第一次迭代中，节点 i 的可能的标记集合 P_i^1 对所有的 i 来说都是 L ，换句话说，所有节点的初始标记是所有可能的标记。在第 k 次迭代过程中，标记算法将从“ P_i^k ”中除去无效标记，得到 P_i^{k+1} 。去除标记的依据建立在节点当前标记、该节点与其它节点的关系、各种约束等基础上，因此，每一个处理器都有足够的信息来独立地对其标记集合 P_i^k 进行细化。这样，所有的处理器就有可能同步工作。需要指出，在任一时刻，处理器只使用能直接可得到的信息，这就是说，只使用从属于该节点对应区域的信息。但是，每一次迭代都通过它的邻节点或关联节点把效应传播给其它没有直接关系的节点，即每一次迭代都会增加节点的影响圈。

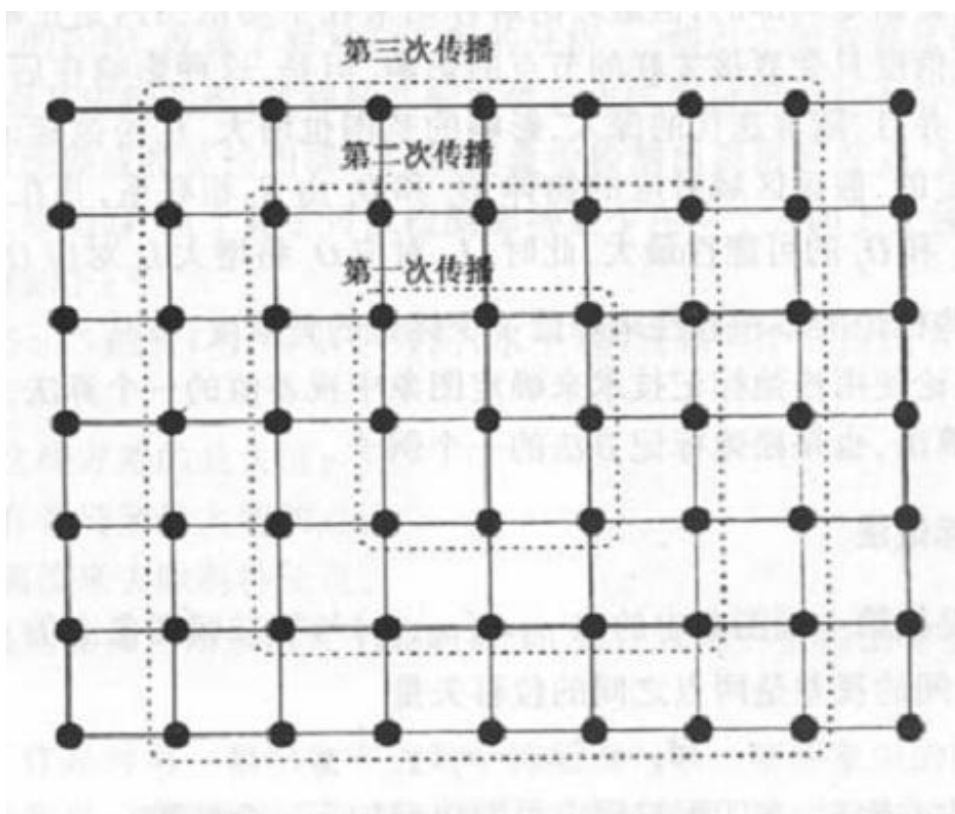


图 14.7 并行传播示意图

对大多数应用来说，在标记过程开始前，有关物体的一些知识是可以得到的。分割或标记前进行的其它处理过程又常常可以提供用于细化节点初始集合 P_i 的知识。标记过程可以进一步细化这些标记集合，以使得每一个区域对应唯一的标记。下面考虑一种与上面所述的标记问题稍微不同的标记问题。基于某种酉关系，标记集合 P_i^1 可以分配给一个区域。为了确定标记分配正确与否，给每个标记 $l_k \in P_i^1$ 确定一个置信度。这个置信度和概率一样，表示了一种信任度，即基于图像提供的证据给某个区域分配某个标记的置信度。因此，对每个元素 $l_k \in P_i$ ，一个非负概率 p_{ik} 表示标记 l_k 是节点 i 的正确标记的置信度。

标记过程实际上是使用约束来细化每个标记的置信度。置信度 p_{ik} 受连通节点标记的置信度影响，这样，在第 t 次迭代中，节点 i 的标记 l_k 的置信度 p_{ik}^t 是置信度 p_{ik}^{t-1} 和所有直接有关的节点标记的置信度函数。在每一次迭代中，一个节点受制于其它所有有关节点的标记，然后使用已知约束来更新该标记的置信度。标记过程的结束有两种情况，一种是当每个节点都有一个唯一的标记，另一种是置信度达到一个稳定的状态。

上面的过程通常称为松弛标记过程，即依据局部的证据决定哪一种可能的解释是正确的。尽管使用的证据是局部的，但最终的解释结果在全局范围内是正确的。在每次迭代中，一个标记的置信度只受直接关联的节点的影响。但是，这种影响在后面的迭代中会传播给其它的节点，并且，随着迭代的深入，影响的范围也增大。在松弛标记中，约束是通过相容性函数来确定的。假设区域对应的物体 O_i 和 O_j 由 R_{ij} 相联系，并在这种联系下标记 L_{i_k} 和 L_{j_l} 分配给 O_i 和 O_j 的可能性最大。此时， L_{i_k} 对应 O_i 将增大 L_{j_l} 对应 O_j 的可能性。同样也有可能利用某些标记的不相容性来降低一个标记的置信度。

下面，我们讨论使用松弛标记技术来确定图像中视差值的一个算法。在本章后面要讨论的确定光流的算法，也是松弛标记方法的一个例子。

(2) 视差计算松弛标记法

匹配问题就是把第一幅图像中的点 $p_i = (x_i, y_i)$ 与第二幅图像上的点 $p_j = (x_j, y_j)$ 对应起来。这两点之间的视差是两点之间的位移矢量：

$$d_{ij} = (x_i - x_j, y_i - y_j) \quad (14.7)$$

匹配的两个点称为共轭对。

在匹配过程中可能用到如下三个性质：

离散性：各点之间明显区别的测度。

相似性：两个点之间相似程度的测度。

一致性：一个匹配点与邻近其它匹配点变化一致程度的测度。

离散性是指特征必须是一个个孤立的点。例如，线段就不是一个好的特征，因为一个点能匹配线段上的许多点。离散性将图像视差分析问题退化成为有限数量点的匹配问题，因此，离散性把搜索的成本降到了最低程度。

潜在匹配点的集合可以形成一个双向联接图，匹配问题就是从中选择一种联接图。最初每个节点都可认为与其它每个节点都有一个匹配联接，如图 14.8 所示。使用某一判据，对应问题的目标就是求每一个节点对应的一个匹配而去除所有其它的联系。相似特性是指两个潜在的匹配点相互接近的程度，这是一个关联性的测度。相似性可以在所选离散特征点的任何性质基础上进行测量。

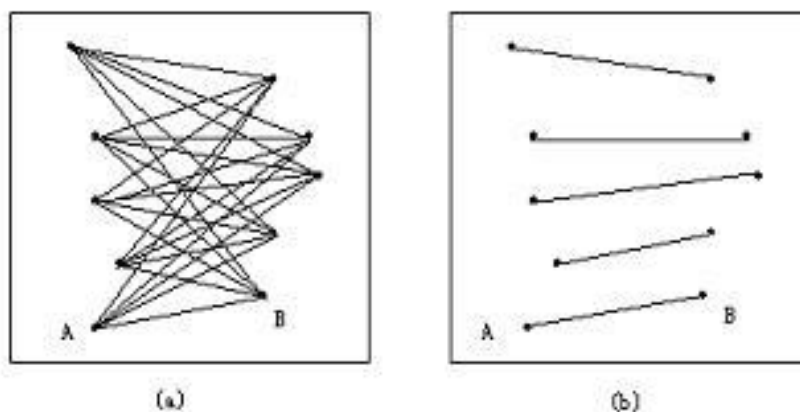


图 14.8 (a)图是一个完全的双向图。A 组的每一个节点与 B 组的每一个节点相联接。使用节点(点)的特征和其它一些知识，对应性算法将给每一个节点只保留一个联接，而消除所有其它的节点，如图(b)所示

假设物体的运动特性良好，则一致性意味着场景中表面的空间连续性。一致性判据可以实现显而易见的匹配，改善了对复杂匹配的分析。一些点之间有着足够的分类特征和相似特征，很容易对其进行匹配；这样的匹配在邻近点匹配过程中是十分有用的。

使用角点检测器或特征检测器可以从图像中检测出离散特征点。Moravec 兴趣算子就是这样一种特征检测器。这个算子可以检测那些至少在一个方向上光强值迅速改变的点。算子执行的步骤如下：

1. 用一个 5×5 的窗口计算四个方向(水平、垂直和两个对角线方向)上像素差平方和。
2. 计算出这些方差的最大值。
3. 抑制所有非局部最大值的点。
4. 用一个阈值来去除弱特征点。

任何特征检测器都可用来取代上面的算子。如计算每一点的曲率值并选择高曲率值作为特征。

接下来的工作是对第一幅图像中的每个特征点与第二幅图像中的所有特征点在最大距离范围内进行配对。这将消除完全双向图中的许多联接。那些消除掉的是两图像中差别很大的点之间的联接，因为它们不可能成为候选匹配点。每一个节点 a_i 包含第一幅

图像中的一个位置 (x_i, y_i) 和一组可能的标记(视差矢量). 一组视差标记是一组位移矢量或是未定义的视差.

匹配的初始概率通过使用两图像特征点之间的相似测度来计算. 人们一般使用对应窗口中所有像素差的平方和 s_i 为测度.

下面的步骤可以用来给这些概率赋值. 设 l 是一个点的一个候选标记, 表示该点的一个视差向量. 我们先计算 $w_i(l)$, 即点 (x_i, y_i) 与其潜在匹配点之间的相似度:

$$w_i(l) = \frac{1}{1 + cs_i(l)} \quad (14.8)$$

式中 $s_i(l)$ 是对应于标记 l 的视差平方和, c 是一个正的常数. 这个点不足以定义视差的概率是:

$$p_i^0 = 1 - \max(w_i(l)) \quad (14.9)$$

这个概率值是由与 (x_i, y_i) 最相似点的相似度来确定的. 如果没有很强的相似点, 那么有可能这个点在这图像中就没有匹配. 各个匹配(标记)点的概率是

$$p_i(l|i) = \frac{w_i(l)}{\sum w_i(l')} \quad (14.10)$$

式中 $p_i(l|i)$ 是条件概率, a_i 节点具有标记 l , 求和是不包含“未定义”标记的所有标记 l' . 使用一致性和迭代松弛算法可以使概率估计逐渐精细. 在这个算法中, 每个节点上的标记是加强还是减弱, 是以本次迭代中相邻节点的标记为基础的. 这里使用的最重要的性质是在给定的邻域中, 所有的视差都是相似的. 这样, 邻域中具有相似视差的节点相互加强, 而视差不相似的节点则被减弱了. 下面的方法可以实现这一思想.

让我们考虑点 a_i 所有邻点的视差矢量概率. 对每个邻点, 将那些与 a_i 的视差相近或相似的标记(视差)概率加起来:

$$q_i^l(l) = \sum_{A_i} \sum_{D_i} p_j^k(l') \quad (14.11)$$

其中 A_i 是 a_i 的所有邻点的集合, D_i 是与 a_i 视差相似的那些标记的集合. 现在使用迭代计算来逐渐精细概率值:

$$p_i^{k+1}(l) = p_i^k(A + Bq_i^k(l)) \quad (14.12)$$

其中常量 A 和 B 用来控制算法的收敛速度. 通常, 只进行几次迭代就能得到一个好的解. 另外, 去掉低概率值的匹配点可以有效地提高算法的速度. 图 14.9 示意的是一个序列中的两幅图像, 图 14.10 是使用上面算法计算视差的迭代过程示意图. 有兴趣的读者可以参见文献[Barbard 1980].

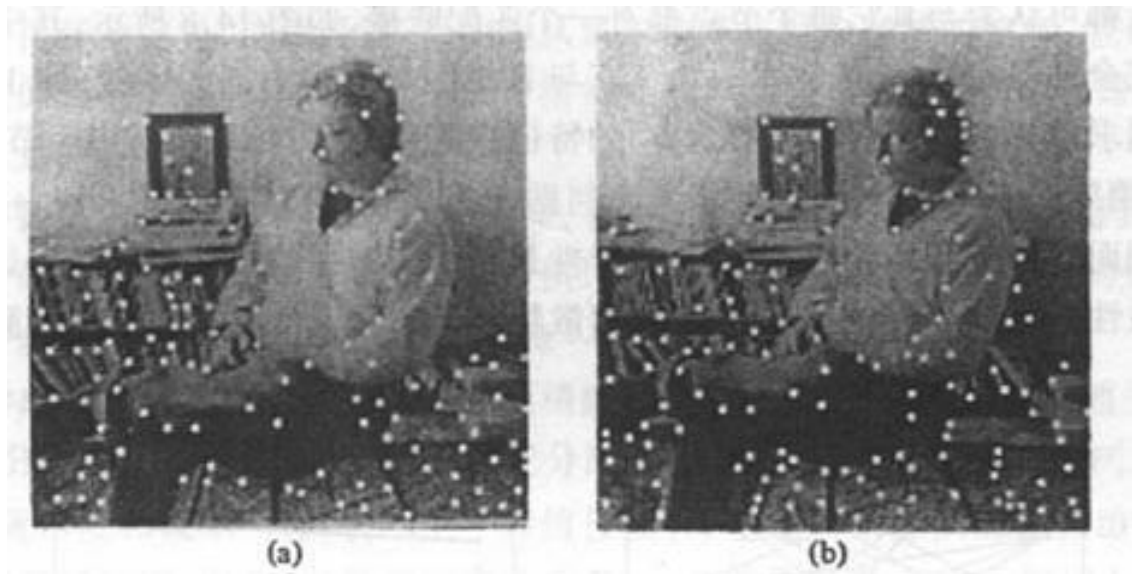


图 14.9 立体图像对

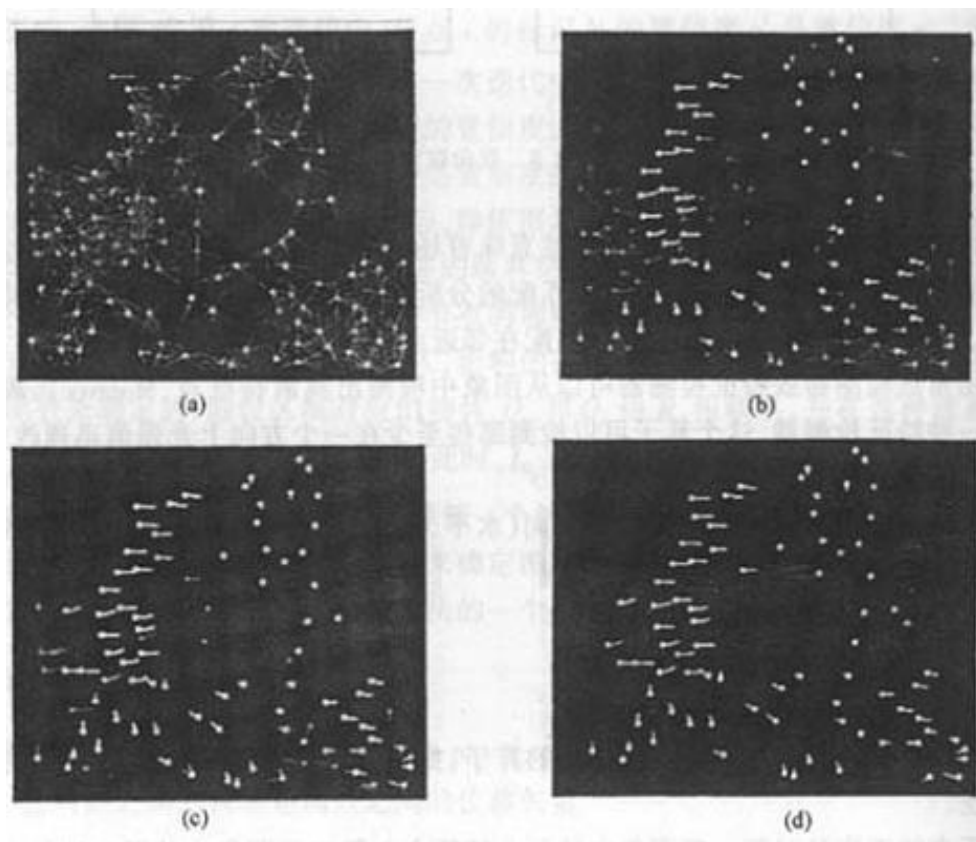


图 14.10 使用松弛算法求匹配特征点视差示意图, (a) 对图 14.9 进行初始概率分配, (b)、(c)和(d)是第 2、第 6 和第 10 次迭代结果

14.2 光流法

光流法是运动图像分析的重要方法. 本节首先介绍光流的基本概念和图像运动估计的基本问题, 然后讨论光流方程, 基于光流方程的一些图像分析方法和算法见 14.3 节。

14.2.1 运动场与光流

给图像中的每一像素点赋予一个速度向量, 就形成了图像运动场(motion field). 在运动的一个特定时刻, 图像上某一点 p_i 对应三维物体上某一点 P_0 , 这种对应关系可以由投影方程得到. 在透视投影情况下, 图像上一点与物体上对应一点的连线经过光学中心, 该连线称为图像点连线(Point ray), 如图 14.11 所示

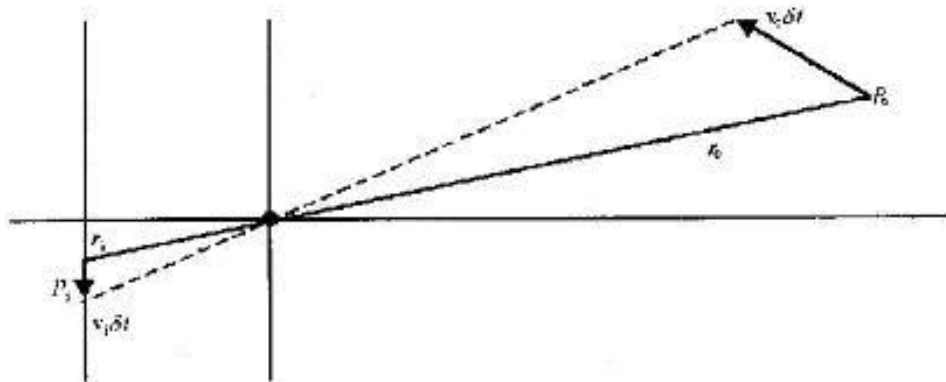


图 14.11 三维物体上一点运动的二维投影

设物体上一点 P_0 相对于摄像机具有速度 \mathbf{v}_0 ，从而在图像平面上对应的投影点 p_i 具有速度 \mathbf{v}_i 。在时间间隔 δt 时，点 P_0 运动了 $\mathbf{v}_0 \delta t$ ，图像点 p_i 运动了 $\mathbf{v}_i \delta t$ 。速度可由下式表示：

$$\mathbf{v}_0 = \frac{d\mathbf{r}_0}{dt} \quad \mathbf{v}_i = \frac{d\mathbf{r}_i}{dt} \quad (14.13)$$

式中 \mathbf{r}_0 和 \mathbf{r}_i 之间的关系为：

$$\frac{1}{f'} \mathbf{r}_i = \frac{1}{\mathbf{r}_0 \cdot \hat{\mathbf{z}}} \mathbf{r}_0 \quad (14.14)$$

其中， f' 表示图像平面到光学中心的距离， $\hat{\mathbf{z}}$ 表示 z 轴的单位矢量。

式 (14.14) 只是用来说明三维物体运动与在图像平面投影之间的关系，但我们关心的是图像亮度的变化，以便从中得到关于场景的信息。

当物体运动时，在图像上对应物体的亮度模式也在运动。光流(optical flow)是指图像亮度模式的表观（或视在）运动 (apparent motion)[Horn 1986]。使用“表观运动”这个概念的主要原因是光流无法由运动图像的局部信息唯一地确定，比如，亮度比较均匀的区域或亮度等值线上的点都无法唯一地确定其点的运动对应性，但运动是可以观察到的。与光流同义的另一个常用术语是图像流(image flow)。

在理想情况下，光流对应于运动场，但这一命题不总是对的。图 14.12 所示的是一个非常均匀的球体，由于球体表面是曲面，因此在某一光源照射下，亮度呈现一定的空间分布或叫明暗模式。当球体在摄像机前面绕中心轴旋转时，明暗模式并不随着表面运动，所以图像也没有变化，此时光流在任意地方都等于零，然而，运动场却不等于零。如果球体不动，而光源运动，明暗模式运动将随着光源运动。此时光流不等于零，但运动场为零，因为物体没有运动。一般情况下可以认为光流与运动场没有太大的区别，因此允许我们根据图像运动来估计相对运动。

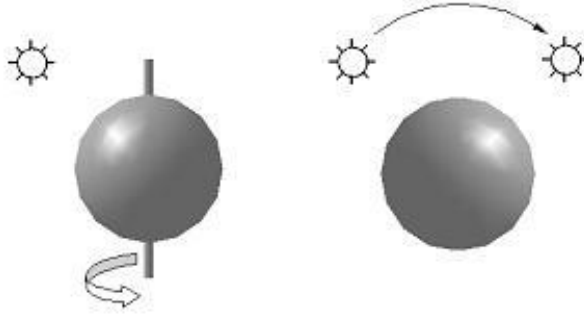


图 14.12 光流与运动场差别示意图

14.2.2 光流约束方程

设 $I(x, y, t)$ 是图像点 (x, y) 在时刻 t 的照度, 如果 $u(x, y)$ 和 $v(x, y)$ 是该点光流的 x 和 y 分量, 假定点在 $t + \delta t$ 时运动到 $(x + \delta x, y + \delta y)$ 时, 照度保持不变, 其中, $\delta x = u \delta t$, $\delta y = v \delta t$, 即

$$I(x + u \delta t, y + v \delta t, t + \delta t) = I(x, y, t) \quad (14.15)$$

这一约束还不能唯一地求解 u 和 v , 因此还需要其它约束, 比如, 运动场处处连续等约束.

如果亮度随 x 、 y 、 t 光滑变化, 则可以将上式的左边用 Taylor 级数展开,

$$I(x, y, t) + \delta x \frac{\partial I}{\partial x} + \delta y \frac{\partial I}{\partial y} + \delta t \frac{\partial I}{\partial t} + e = I(x, y, t) \quad (14.16)$$

其中 e 是关于 δx 、 δy 、 δt 的二阶和二阶以上的项. 上式两边的 $I(x, y, t)$ 相互抵消, 两边除以 δt , 并取极限 $\delta t \rightarrow 0$, 得到

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \quad (14.17)$$

上式实际上是下式的展开式

$$\frac{dI}{dt} = 0 \quad (14.18)$$

设

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y} \quad I_t = \frac{\partial I}{\partial t}$$

$$u = \frac{dx}{dt} \quad v = \frac{dy}{dt}$$

则由式(14.17)得到空间和时间梯度与速度分量之间的关系:

$$I_x u + I_y v + I_t = 0 \quad (14.19)$$

或

$$\nabla \mathbf{I} \cdot \mathbf{v} + I_t = 0 \quad (14.20)$$

上述方程称为光流约束方程. 在上面的方程中, I_x , I_y 和 I_t 可直接从图像中计算出来.

实际上,上述光流约束方程产生的是恒值亮度轮廓图像运动的法向分量 $\mathbf{v}_n = s\mathbf{n}$, 其中 \mathbf{n} 和 s 分别是法向运动分量的方向和大小:

$$\mathbf{n} = \frac{\nabla \mathbf{I}}{\|\nabla \mathbf{I}\|} \quad s = \frac{-I_t}{\|\nabla \mathbf{I}\|} \quad (14.21)$$

图像中的每一点上有两个未知数 u 和 v , 但只有一个方程, 因此, 只使用一个点上的信息是不能确定光流的. 人们将这种不确定问题称为孔径问题(aperture problem). 理论上分析, 我们仅能沿着梯度方向确定图像点的运动, 即法向流(normal flow). 假定物体的运动方向为 \mathbf{r} , 如图 14.12 所示. 如果基于一个局部窗口(即孔径 1)来估计运动, 则无法确定图像是沿着边缘方向还是垂直边缘方向运动, 其中沿着垂直边缘方向的运动就是法向流. 但是, 如果我们再来观察孔径 2, 就有可能确定正确的运动, 这是由于图像在孔径 2 中的两个垂直边缘方向上都有梯度变化. 这样, 在一个包含有足够灰度变化的像素块上有可能估计图像运动. 当然, 这里隐含着一个假设, 那就是像素块里的所有像素都具有相同的运动矢量.

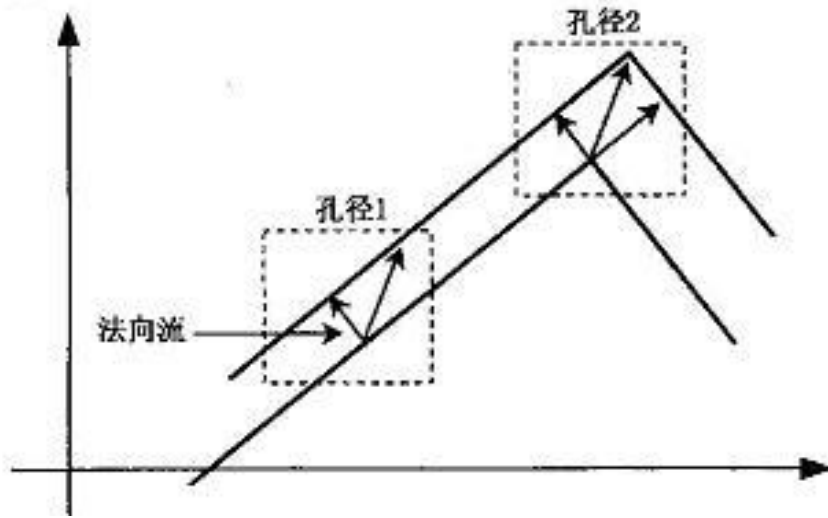


图 14.13 孔径问题示意图

14.3 光流计算

由上节讨论可知, 由于孔径问题的存在, 仅通过光流约束方程而不使用其它信息是无法计算图像平面中某一点处的图像流速度, 本节将讨论如何克服孔径问题, 并求出图像流的几种方法。

14.3.1 Horn-Schunck 法

Horn 和 Schunck[Horn]使用光流在整个图像上光滑变化的假设来求解光流, 即运动场既满足光流约束方程又满足全局平滑性. 根据光流约束方程, 光流误差为

$$e^2(\mathbf{x}) = (I_x u + I_y v + I_t)^2 \quad (14.22)$$

其中 $\mathbf{x} = (x, y)^T$. 对于光滑变化的光流, 其速度分量平方和积分为

$$s^2(\mathbf{x}) = \iint \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right] dx dy \quad (14.23)$$

将光滑性测度同加权微分约束测量组合起来, 其中加权参数控制图像流约束微分和光滑性微分之间的平衡:

$$E = \iint \{e^2(\mathbf{x}) + \alpha s^2(\mathbf{x})\} dxdy \quad (14.24)$$

其中 α 是控制平滑度的参数, α 越大, 则平滑度就越高, 则估计的精度也越高. 使用变分法将上式转化为一对偏微分方程:

$$\begin{aligned} \alpha \nabla^2 u &= I_x^2 u + I_x I_y v + I_x I_t \\ \alpha \nabla^2 v &= I_x I_y u + I_y^2 v + I_y I_t \end{aligned} \quad (14.25)$$

用有限差分方法将每个方程中的拉普拉斯算子换成局部邻域图像流矢量的加权和, 并使用迭代方法求解这两个差分方程.

下面只考虑离散的情况. 在一点 (i, j) 及其 4 邻域上, 根据光流约束方程, 光流误差的离散量表示式为:

$$e^2(i, j) = (I_x u(i, j) + I_y v(i, j) + I_t)^2 \quad (14.26)$$

光流的平滑量也可由点 (i, j) 与它的 4 邻域点的光流值差分来计算:

$$\begin{aligned} s^2(i, j) &= \frac{1}{4} [(u(i, j) - u(i-1, j))^2 + (u(i+1, j) - u(i, j))^2 \\ &\quad + (u(i, j+1) - u(i, j))^2 + (u(i, j) - u(i, j-1))^2 \\ &\quad + (v(i, j) - v(i-1, j))^2 + (v(i+1, j) - v(i, j))^2 \\ &\quad + (v(i, j+1) - v(i, j))^2 + (v(i, j) - v(i, j-1))^2] \end{aligned} \quad (14.27)$$

则极小化函数为:

$$E = \sum_i \sum_j (e^2(i, j) + \alpha s^2(i, j)) \quad (14.28)$$

E 关于 u 和 v 的微分是:

$$\begin{aligned} \frac{\partial E}{\partial u} &= 2(I_x u + I_y v + I_t) I_x + 2\alpha(u - \bar{u}) \\ \frac{\partial E}{\partial v} &= 2(I_x u + I_y v + I_t) I_y + 2\alpha(v - \bar{v}) \end{aligned} \quad (14.29)$$

其中 \bar{u} 和 \bar{v} 分别是 u 和 v 在点 (i, j) 处的平均值. 当上式为零时, 则式 14.28 取极小值, 因此得到:

$$\begin{aligned} (I_x u + I_y v + I_t) I_x + \alpha(u - \bar{u}) &= 0 \\ (I_x u + I_y v + I_t) I_y + \alpha(v - \bar{v}) &= 0 \end{aligned} \quad (14.30)$$

从上面两个方程可以求出 u 和 v . 实际中, 经常将求解 u 和 v 表示成迭代方程:

$$\begin{aligned} u^{n+1} &= \bar{u}^n - I_x \frac{I_x \bar{u}^n + I_y \bar{v}^n + I_t}{\alpha + I_x^2 + I_y^2} \\ v^{n+1} &= \bar{v}^n - I_y \frac{I_x \bar{u}^n + I_y \bar{v}^n + I_t}{\alpha + I_x^2 + I_y^2} \end{aligned} \quad (14.31)$$

其中 n 是迭代次数, u^0 和 v^0 是光流的初始估值, 一般取为零. 当相邻两次迭代结果的距离小于预定的公差值, 迭代过程终止. Horn-Schunck 光流法实验结果见图 14.15.

14.3.2 Lucas-Kanade 方法

Lucas和Kanade[Lucas 1981] 假设在一个小的空间邻域 Ω 上运动矢量保持恒定, 然后使用加权最小二乘方 (weighted least-squares) 估计光流. 在一个小的空间邻域 Ω 上, 光流估计误差定义为

$$\sum_{(x,y) \in \Omega} W^2(\mathbf{x}) (I_x u + I_y v + I_t)^2 \quad (14.32)$$

其中 $W(\mathbf{x})$ 表示窗口权重函数，它使邻域中心部分对约束产生的影响比外围部分更大。设 $\mathbf{v} = (u, v)^T$ ， $\nabla \mathbf{I}(\mathbf{x}) = (I_x, I_y)^T$ 式(14.32)的解由下式给出：

$$\mathbf{A}^T \mathbf{W}^2 \mathbf{A} \mathbf{v} = \mathbf{A}^T \mathbf{W}^2 \mathbf{b} \quad (14.33)$$

其中，在时刻 t 的 n 个点 $\mathbf{x}_i \in \Omega$ ，

$$\begin{aligned} \mathbf{A} &= [\nabla \mathbf{I}(\mathbf{x}_1), \dots, \nabla \mathbf{I}(\mathbf{x}_n)]^T, \\ \mathbf{W} &= \text{diag}[W(\mathbf{x}_1), \dots, W(\mathbf{x}_n)], \\ \mathbf{b} &= -(I_t(\mathbf{x}_1), \dots, I_t(\mathbf{x}_n))^T. \end{aligned}$$

式(14.33)的解为 $\mathbf{v} = [\mathbf{A}^T \mathbf{W}^2 \mathbf{A}]^{-1} \mathbf{A}^T \mathbf{W}^2 \mathbf{b}$ ，其中当 $\mathbf{A}^T \mathbf{W}^2 \mathbf{A}$ 为非奇异时可得到解析解，因为它是一个 2×2 的矩阵：

$$\mathbf{A}^T \mathbf{W}^2 \mathbf{A} = \begin{bmatrix} \sum W^2(\mathbf{x}) I_x^2(\mathbf{x}) & \sum W^2(\mathbf{x}) I_x(\mathbf{x}) I_y(\mathbf{x}) \\ \sum W^2(\mathbf{x}) I_y(\mathbf{x}) I_x(\mathbf{x}) & \sum W^2(\mathbf{x}) I_y^2(\mathbf{x}) \end{bmatrix} \quad (14.34)$$

其中所有的和都是在邻域 Ω 上的点得到的。

等式(14.32)和(14.33)也可认为是从法向速度(normal velocities) $\mathbf{v}_n = \mathbf{s} \mathbf{n}$ 得到的估计 \mathbf{v} 的加权最小二乘估计(weighted least-squares estimates)；即(14.32)等于：

$$\sum_{\mathbf{x} \in \Omega} W^2(\mathbf{x}) w^2(\mathbf{x}) [\mathbf{v} \cdot \mathbf{n}(\mathbf{x}) - s(\mathbf{x})]^2 \quad (14.35)$$

我们的实现首先用标准差为1.5像素/帧的时空高斯滤波器平滑图象序列。这有助与削弱时间噪声(temporal aliasing)和输入中的量化效应(quantization effects)。梯度的计算使用了4点中心差，其系数模板为 $(-1 \ 8 \ 0 \ -8 \ 1)/12$ 。空间邻域 Ω 为 5×5 像素大小，窗口权重函数 $W^2(\mathbf{x})$ 为可分离的和各向同性的；在[Simoncelli 1991]中的有效的一维权重为

$(0.0625 \ 0.25 \ 0.375 \ 0.25 \ 0.0625)$ 。在整个处理中需要15帧图象。Fleet and Langley [Fleet 1993]已经用IIR递归滤波器和时间上的递归估计替代了FIR滤波器。这种方法只需存储3帧图象（即只有2到3帧的延迟）就可产生相似的结果。

Simoncelli 等在[Simoncelli 1991]中提出了一个(14.32)的贝叶斯透视图(Bayesian perspective)。他们在梯度测量中使用高斯分布误差(Gaussian distributed errors)和高斯分布的先验速度 \mathbf{V} 来模拟光流约束方程(14.17)。最大化一个后验解的结果与式(14.33)相似，并且产生一个速度估计的协方差矩阵。我们发现这种修改不会明显的改变准确性，但它提出了使用 $\mathbf{A}^T \mathbf{W}^2 \mathbf{A}$ ($\lambda_1 \geq \lambda_2$) 的特征值来鉴别不可靠的估计，该值依赖于空间梯度的数量和方向的范围。虽然 Simoncelli 等建议使用特征值的和，但我们发现单个的最小特征值更加可靠一些。因此在我们的实现中，如果 λ_1 和 λ_2 都大于一个阈值 τ ， \mathbf{V} 用(14.33)计算；如果 $\lambda_1 \geq \tau$ 而 $\lambda_2 < \tau$ ，计算法向速度估计；如果 $\lambda_1 < \tau$ ，不计算速度。

14.3.3 Nagel 方法

Nagel使用二阶导数(second-order derivatives)来估计光流[Nagel 1983, 1987, 1989]。和Horn-Schunck法一样，Nagel也使用了全局平滑约束来建立光流误差测度函数，与Horn-Schunck 测度函数(14.24)不同，Nagel提出的一种面向平滑的约束

(oriented-smoothness constraint)，并不是强加在亮度梯度变化最剧烈的方向（即边缘方向）上，这样做的目的是为了处理遮挡(occlusion)问题[Nagel 1986]。该方法的误差测度函数为

$$\begin{aligned} & \iint (I_x u + I_y v + I_t)^2 + \frac{\alpha^2}{\|\nabla \mathbf{I}\|_2^2 + 2\delta} [(u_x I_y - u_y I_x)^2 \\ & + (v_x I_y - v_y I_x)^2 + \delta(u_x^2 + u_y^2 + v_x^2 + v_y^2)] dx dy \end{aligned} \quad (14.36)$$

相对于 \mathbf{v} 求上式的极小化会削弱垂直于梯度方向上的光流变化。[Nagel 1987]建议取 $\delta=1.0$, $\alpha=0.5$ 。

使用 Gauss-Seidel 迭代, (14.36)的解可表示为:

$$\begin{aligned} u^{k+1} &= \xi(u^k) - \frac{I_x[I_x\xi(u^k) + I_y\xi(v^k) + I_t]}{I_x^2 + I_y^2 + \alpha^2} \\ v^{k+1} &= \xi(v^k) - \frac{I_y[I_x\xi(u^k) + I_y\xi(v^k) + I_t]}{I_x^2 + I_y^2 + \alpha^2} \end{aligned} \quad (14.37)$$

其中, k 表示迭代次数, $\xi(u^k)$ 和 $\xi(v^k)$ 由下式给出:

$$\begin{aligned} \xi(u^k) &= \bar{u}^k - 2I_xI_yu_{xy} - \mathbf{q}^T(\nabla u^k) \\ \xi(v^k) &= \bar{v}^k - 2I_xI_yv_{xy} - \mathbf{q}^T(\nabla v^k) \end{aligned} \quad (14.38)$$

其中:

$$\mathbf{q} = \frac{1}{I_x^2 + I_y^2 + 2\sigma} \nabla \mathbf{I}^T \left[\begin{pmatrix} I_{yy} & -I_{xy} \\ -I_{xy} & I_{xx} \end{pmatrix} + 2 \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix} \mathbf{W} \right]$$

u_{xy}^k 和 v_{xy}^k 表示 \mathbf{v}^k 的偏导数的估计, \bar{u}^k 和 \bar{v}^k 是 u^k 和 v^k 的局部邻域的平均, \mathbf{W} 加权矩阵:

$$\mathbf{W} = (I_x^2 + I_y^2 + 2\delta)^{-1} \begin{pmatrix} I_y^2 + \delta & -I_xI_y \\ -I_xI_y & I_x^2 - \delta \end{pmatrix}$$

在实现中,所有的速度初值都可设为0。图象序列使用了一个在时空上标准差均为1.5像素的高斯核进行滤波预处理。亮度导数使用4点中心差算子计算,在不同的方向上层叠以得到另一个导数(second derivatives)。一阶速度导数用2点中心差核(0.5,0,-0.5)计算,而二阶导数通过层叠一阶导数计算而得。图14.15所示的是使用了100次迭代得到的光流结果结果。关于实现的细节可参考[Barron 1993]。



第1帧

第20帧

(a) 第一帧

(b) 第二十帧

图 14.14 兵马俑模型图像序列



图 14.15 光流实验结果, (a) Horn-Schunck 法, (b) Lucas-Kanade 法, (c) Nagel 法

14.3.4 鲁棒计算方法

显然, 光流约束方程 14.21 不适用遮挡背景的运动过程, 也就是说, 在运动边界处的运动信息是不可靠的, 即可能产生不正确的运动点或局外点. 鲁棒计算方法可以避免边界处不正确运动约束带来的问题.

图像流可用最小中值二乘回归法这一鲁棒回归算法来计算. 将最小中值二乘法作用于连接的各个邻域. 在每一个邻域里, 该算法将尽可能找出所有可能的约束线对. 计算每一对约束线的交点, 并计算残差平方的中值, 以便给每一个估计值指定一个权值. 然后, 将每一个交点及其权值都存储起来. 在试过所有的约束线对以后, 对应于最小权值的交点作为邻域中心图像流速度的估计.

使用约束线在速度空间的距离 d 和图像梯度角 α , 可将约束线表示为极坐标形式:

$$d = \rho \cos(\alpha - \beta) \quad (14.39)$$

式中 $\rho(x, y)$ 和 $\beta(x, y)$ 分别表示运动的速度大小和方向. 设第一条约束线的坐标为 d_1 和 α_1 , 第二条约束线的坐标为 d_2 和 α_2 . 在直角坐标系中交点的位置是

$$\begin{aligned} x &= \frac{d_1 \sin \alpha_2 - d_2 \sin \alpha_1}{\sin(\alpha_1 - \alpha_2)} \\ y &= \frac{d_2 \cos \alpha_1 - d_1 \cos \alpha_2}{\sin(\alpha_1 - \alpha_2)} \end{aligned} \quad (14.40)$$

该模型对约束线的拟合度是残差平方的中值:

$$\text{med}_i(r_i^2) \quad (14.41)$$

运动估计和每条约束线之间的残差 r 是约束线与估值点 (x,y) 间的垂直距离. 残差由下式给出

$$r = x \cos \alpha + y \sin \alpha \quad (14.42)$$

由方程(14.40) 给出的约束线对的交点位置是一个候选解. 计算并存储相对于候选解的约束线残差平方的中值, 连同候选解一起作为一个潜在解. 残差平方的中值是邻域中每一条线与候选交点间垂直距离的平方中值.

典型的邻域尺寸是 5×5 . 一个 $n \times n$ 的邻域包含了 n^2 条约束线. 那么 $n \times n$ 邻域里可能的约束线对数是

$$\frac{n^2(n^2 - 1)}{2} \quad (14.43)$$

一个 5×5 邻域会产生 300 对线. 如果计算时间有限, 则没有必要试验所有的线对. Rousseeuw 和 Leroy [Rousseeuw 1987, p198] 提供了一个试验次数表, 给出在 95% 置信度下, 用 p 个参数的模型拟合含有不同比例局外点的数据集的最小实验次数, 其中假定邻域里至多有 50% 的局外点. 图像流速度场的局部估计仅需要两条约束线. 从 Rousseeuw 和 Leroy 给出的表中可以看到, 只需要试验 11 对约束线就能提供具有 95% 置信度的一致性估计. 使用更多的线对可能会给计算一致性估计的增加麻烦. 如果使用的线对数量少于所有可能的线对, 则必须对线对进行挑选, 以使每对线中的约束离的越远越好. 这就减少了因求解相近方向约束线交点带来的病态问题. 在每一个邻域里应该使用一个预编程方案, 以便选择约束线对. 这种方法的实验结果如图 14.16 所示.

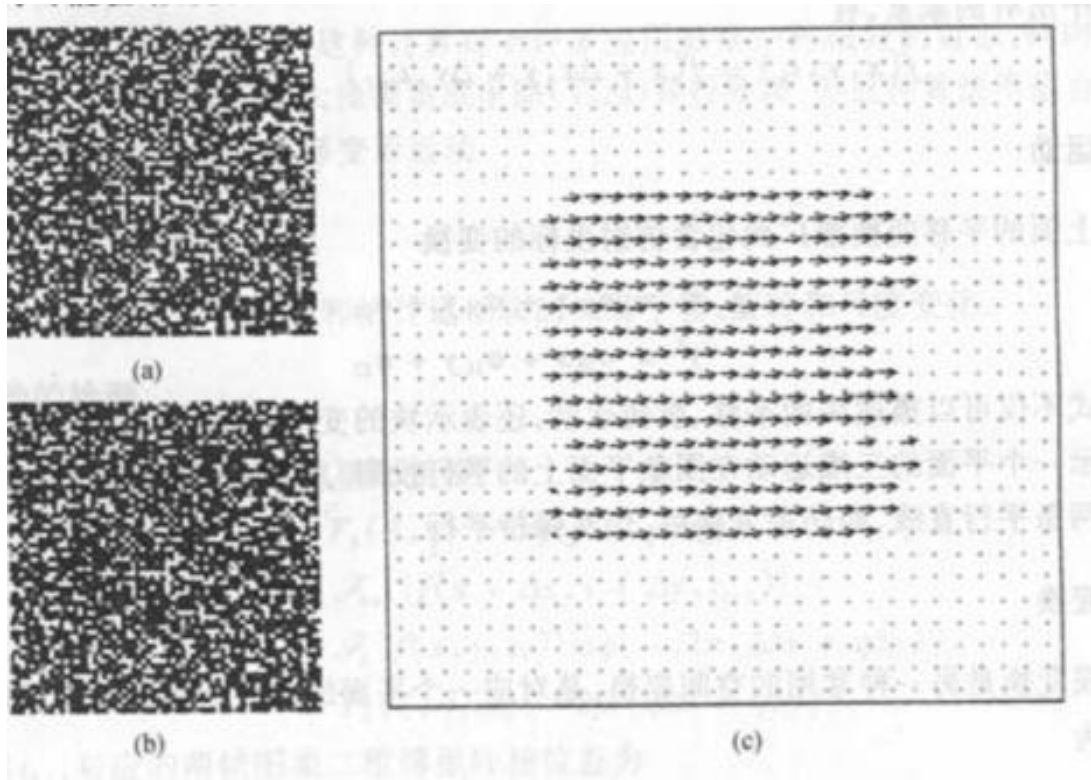


图 14.16 使用最小中值二乘算法的进行光流计算的实验结果

14.4 基于块的运动分析

基于块(Block-based))的运动分析在图像运动估计和其它图像处理和分析中得到了广泛的应用, 比如在数字视频压缩技术中, 国际标准 MPEG1-2 采用了基于块的运动分析和补偿

算法。块运动估计与光流计算不同，它无需计算每一个像素的运动，而只是计算由若干像素组成的像素块的运动，对于许多图像分析和估计应用来说，块运动分析是一种很好的近似。

14.4.1 块运动模型

基于块的运动模型假设图像运动可以用块运动来表征。块运动通常分为平移、旋转、仿射、透视等运动形式，一般情况下，块运动是这些运动的组合，称为变形运动(deformation motion)。

(1) 平移运动

假设图像中每一个块都作平移运动。在第 k 帧图像中选取一个块 B ，其中心位于 (x, y) 。在第 $k+1$ 帧时，块 B 的所有像素之间的关系及其灰度值保持不变，但块中心运动到 (x', y') ，将第 k 帧到第 $k+1$ 帧的平移变换公式表示为

$$\begin{aligned} x' &= x + \Delta x \\ y' &= y + \Delta y \end{aligned} \quad (14.44)$$

则对块中所有的像素，有

$$f(x, y, t_k) = f(x + \Delta x, y + \Delta y, t_{k+1}) \quad (x, y) \in B \quad (14.45)$$

(2) 仿射运动

将上面的平移变换推广到包含仿射坐标的变换：

$$\begin{aligned} x' &= a_{00}x + a_{01}y + a_{02} \\ y' &= a_{10}x + a_{11}y + a_{12} \end{aligned} \quad (14.46)$$

该式不仅可以描述块的平移、旋转运动，还表示块的变形运动如图 14.17 所示。仿射变换表示一个平面的三维运动在图像平面上的平行投影，它有一个重要的性质就是平面上任意两条直线，经仿射变换后，仍然保持平行。

(3) 透视投影变换

$$\begin{aligned} x' &= \frac{a_{00}x + a_{01}y + a_{02}}{a_{20}x + a_{21}y + 1} \\ y' &= \frac{a_{10}x + a_{11}y + a_{12}}{a_{20}x + a_{21}y + 1} \end{aligned} \quad (14.47)$$

(4) 二次线性变换形式

$$\begin{aligned} x' &= a_{00}x + a_{01}y + a_{02}xy + a_{03} \\ y' &= a_{10}x + a_{11}y + a_{12}xy + a_{13} \end{aligned} \quad (14.48)$$

这些方法可以解决光流的孔径问题。

块的平移运动具有广泛的应用，主要原因是它无需向光流那样，计算每一个像素的运动，而是只计算一个块的运动，这种计算对于许多应用都是一种很好的近似，同时计算极其简单和有效，非常适合于 VLSI 并行处理。但这种算法不适合旋转运动、图像的缩放运动和图像局部变形运动。

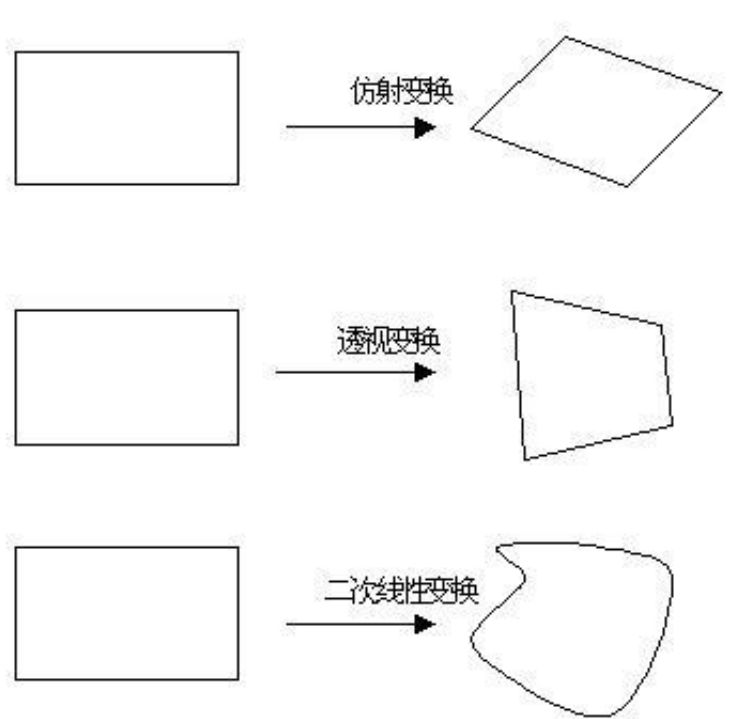


图 14.17 几种空间变换示意图

14.4.2 傅里叶方法

采用傅里叶变换可以检测和估计运动块的二维平移、旋转和尺度变化。

(1) 平移运动的检测

对方程(14.45)的两边取傅里叶变换：

$$F_k(\xi, \eta) = \Phi_k\{f(x, y, t_k)\} \quad (14.49)$$

$$\begin{aligned} F_{k+1}(\xi, \eta) &= \Phi_{k+1}\{f(x + \Delta x, y + \Delta y, t_{k+1})\} \\ &= \Phi_k\{f(x, y, t_k)\} \exp[-j2\pi(\xi\Delta x + \eta\Delta y)] \\ &= F_k(\xi, \eta) \exp[-2j\pi(\xi\Delta x + \eta\Delta y)] \end{aligned} \quad (14.50)$$

时刻 t_k 和 t_{k+1} 对应的两帧图像二维傅里叶相位差为

$$\begin{aligned} \Delta\varphi(\xi, \eta) &= \angle F_{k+1}(\xi, \eta) - \angle F_k(\xi, \eta) \\ &= -2\pi(\xi\Delta x + \eta\Delta y) \end{aligned} \quad (14.51)$$

上式实际上是一个由两个变量 (ξ, η) 定义的一个平面。所以估计块的运动矢量就是估计该平面的法线方向 $(\Delta x, \Delta y)$ 。下面介绍一种求解 $(\Delta x, \Delta y)$ 的直接方法。由变量的可分离性， $f_x(x, t_k)$ 和 $f_x(x, t_{k+1})$ 分别表示 $f(x, y, t_k)$ 和 $f(x, y, t_{k+1})$ 在 x 轴的投影，即

$$\begin{aligned} f_x(x, t_k) &= \sum_y f(x, y, t_k) \\ f_x(x, t_{k+1}) &= \sum_y f(x, y, t_{k+1}) \end{aligned} \quad (14.52)$$

$F_k(\xi, \eta)$ 和 $F_{k+1}(\xi, \eta)$ 分别为 $f_x(x, t_k)$ 和 $f_x(x, t_{k+1})$ 的傅里叶变换，则它们的相位差为：

$$\begin{aligned} \Delta\varphi_x(\xi) &= \angle F_{k+1}(\xi) - \angle F_k(\xi) \\ &= -2\pi\xi\Delta x \end{aligned} \quad (14.53)$$

在频率域，可以得到关于 ξ 的 $\Delta\varphi(\xi)$ ，从上式可解得 Δx

$$\Delta x = -\frac{\Delta \varphi_x(\xi)}{2\pi\xi} \quad (14.54)$$

同理可以求出 Δy ：

$$\Delta y = -\frac{\Delta \varphi_y(\eta)}{2\pi\eta} \quad (14.55)$$

这样 $(\Delta x, \Delta y)$ 就是时刻 t_k 到 t_{k+1} 图像块运动的位移量。

上述直接方法有两个假设：

1. 二维相位函数(14.51)是可以展开的
2. 所选的图像块内只包含一个物体。

第一个假设在实际中一般很难做到。对于第二个假设,如果机器自动地在图像上选择运动块,如果块中包含有两个物体,则估计结果将是错误的。14.4.3 将介绍的相位相关法可以有效地解决上述两个问题。

(2) 旋转运动和尺度变化的检测

由傅立叶方法可知,对于图像的空间域线性模式,如直的边缘,经傅立叶变换后,它的功率谱也呈线性分布,且通过 (ξ, η) 平面的原点,它的方向与图像空间域上的线性模式方向正交。这样当线条旋转时,其线性功率谱也做相应的旋转,由此求出二维旋转运动。如果物体图像的尺度变大,反映在傅立叶频率域上的功率谱的低频分量增大,反之,若物体区域尺寸变小,那么频率域上的功率谱的高频分量增大。这样可以根据功率谱的变化描述物体尺度的变化。

14.4.3 相位相关法

图像序列中的 t_k 和 t_{k+1} 时刻的图像之间互相关函数为

$$c_{k,k+1}(x, y) = f(x, y, t_k) * f(x, y, t_{k+1}) \quad (14.56)$$

对上式两边做傅里叶变换,得到互功率谱为:

$$C_{k,k+1}(\xi, \eta) = \Phi_k(\xi, \eta) \bar{\Phi}_{k+1}(\xi, \eta) \quad (14.57)$$

上式除以互功率谱幅值得到标准互功率谱

$$\tilde{C}_{k,k+1}(\xi, \eta) = \frac{\Phi_k(\xi, \eta) \bar{\Phi}_{k+1}(\xi, \eta)}{\|\Phi_k(\xi, \eta) \bar{\Phi}_{k+1}(\xi, \eta)\|} \quad (14.58)$$

将 14.51 代入上式,则有

$$\tilde{C}_{k,k+1}(\xi, \eta) = \exp[-2j\pi(\xi\Delta x + \eta\Delta y)] \quad (14.59)$$

对上式取傅里叶反变换,得到相位相关函数表达式:

$$\tilde{c}_{k,k+1}(x, y) = \delta(x - \xi, y - \eta) \quad (14.60)$$

由上式可见,相位相关函数是一个脉冲函数,其脉冲位置就是块位移矢量。

求解位移矢量算法见算法 14.1。用计算机实现上述算法时,要用二维离散傅里叶变换(DFT)替换二维傅里叶变换。

算法 14.1 相位相关法运动矢量算法

1. 计算第 t_k 和 t_{k+1} 时刻对应的图像中块的二维 DFT。
2. 根据式(14.58)计算互功率谱相位。
3. 计算 $\tilde{C}_{k,k+1}(\xi, \eta)$ 的逆的二维 DFT, 得到 $\tilde{c}_{k,k+1}(x, y)$ 。
4. 检测相位函数的峰值位置。

理想情况下得到的相位相关函数只有一个峰值,对应着块在两帧图像上的相对位移。但在实际中,许多因素常常会使相位相关函数恶化,从而有可能产生多个峰值。这些因素包括使用二维 DFT 替换二维傅里叶变换,一个块内有两个运动物体,或图像噪声等。

用二维 DFT 替换二维傅里叶变换会产生如下问题：

1. 边界效应 为了得到完整的脉冲函数，移位必须是周期的。由于在窗口的一端消失的像素在窗口的另一端不会再出现，因此，脉冲函数将退化为波峰函数。这就是为什么二维 DFT 算法假设在窗口两端方向上具有周期性。不论是从左到右，还是从上到下，如果存在不连续的边界，都会引进假的峰值。
2. 频谱泄漏： 由于运动矢量为非整数，因而造成频谱泄漏。为了得到完整的脉冲函数，位移矢量的分量必须对应一个整数倍的基频。否则，脉冲函数将退还成波峰函数。
3. 位移估计范围： 由于二维 DFT 是块尺寸大小 (m, n) 的周期函数，因此需要校正位移估计值，以便适合负的位移

$$(\Delta\tilde{x}, \Delta\tilde{y}) = \begin{cases} (\Delta x, \Delta y) & \text{如果 } |\Delta x| \leq (m-1)/2, |\Delta y| \leq (n-1)/2, m, n \text{ 是奇数} \\ (\Delta x - m, \Delta y - n) & \text{其它} \end{cases}$$

这样估计的范围为由点 $(-(m-1)/2, -(n-1)/2)$ 到点 $((m-1)/2, (n-1)/2)$ 组成的块，比如估计的位移范围为 $(-25, -25)$ 到 $(25, 25)$ ，则块的尺寸至少为 $(51, 51)$ 。

块的尺寸选择对于基于块的运动算法是非常重要的。选择较大的块尺寸，可以跟踪大位移的运动。但块尺寸选得太大，块内的所有运动矢量很难保证完全一样，造成估计误差太大。因此在选择窗口尺寸上有一个最佳的折衷方案。使用多分辨率图像结构可以在一定程度上解决这一问题。

14.4.4 块匹配方法

块匹配算法的基本思想如图 14.18 所示。在第 k 帧中选择以 (x, y) 为中心、大小为 $m \times n$ 的块 W ，然后在第 $k+1$ 帧中的一个较大的搜索窗口内寻找与块 W 尺寸相同的最佳匹配块的中心的位移矢量 $\mathbf{r} = (\Delta x, \Delta y)$ 。搜索窗口一般是以第 k 帧中的块 W 为中心的一个对称窗口，其大小常常根据先验知识或经验来确定。各种块匹配算法的差异主要体现在如下几个方面：

- 匹配准则
- 搜索策略
- 块尺寸选择方法

下面介绍几种方法

(1) 匹配准则

典型的匹配准则有：最大互相关准则，最小均方差准则，最小平均绝对值差，最大匹配像素数量准则等。其中最大互相关准则见 14.4.3 节。

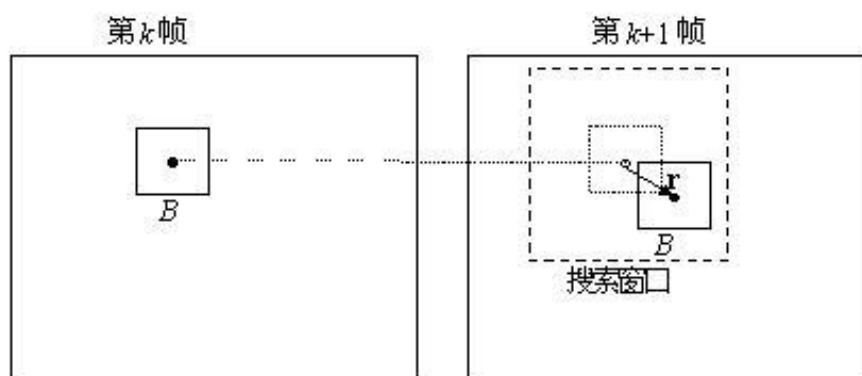


图 14.18 块匹配示意图

最小均方差准则(mean square error, MSE)定义如下：

$$MSE(\Delta x, \Delta y) = \frac{1}{mn} \sum_{(x,y) \in W} [I(x, y, k) - I(x + \Delta x, y + \Delta y, k + 1)]^2 \quad (14.61)$$

通过求上式的极小化可以估计出位移矢量 $\mathbf{r} = (\Delta x, \Delta y)$ ，即

$$[\Delta x, \Delta y]^T = \arg \min_{(\Delta x, \Delta y)} MSE(\Delta x, \Delta y) \quad (14.62)$$

对 MSE 求极小化的准则可以认为是给窗口内的所有像素强加一个光流约束。最小均方差准则很少通过超大规模集成电路(VLSI)来实现，主要原因是用硬件实现平方运算有相当的困难。通过超大规模集成电路(VLSI)来实现的准则是最小平均绝对差。

最小平均绝对差准则(mean absolute difference, MAD)定义如下

$$MAD(\Delta x, \Delta y) = \frac{1}{mn} \sum_{(x,y) \in W} |I(x, y, k) - I(x + \Delta x, y + \Delta y, k + 1)| \quad (14.63)$$

位移矢量 $\mathbf{r} = (\Delta x, \Delta y)$ 的估计值为

$$[\Delta x, \Delta y]^T = \arg \min_{(\Delta x, \Delta y)} MAD(\Delta x, \Delta y) \quad (16.64)$$

众所周知，随着搜索区域的扩大，出现多个局部极小值的可能性也增大，此时，MAD 准则性能将恶化。

还有一种匹配准则是最大匹配像素数量准则(matching pel count, MPC)，这种方法是将窗口内的匹配像素和非匹配像素根据下式分类：

$$T(x, y, \Delta x, \Delta y) = \begin{cases} 1 & \text{如果 } |I(x, y, k) - I(x + \Delta x, y + \Delta y, k + 1)| \leq T \\ 0 & \text{其它} \end{cases} \quad (16.65)$$

T 是预先确定的阈值。这样，最大匹配像素数量准则为

$$MPC(\Delta x, \Delta y) = \sum_{(x,y) \in W} T(x + \Delta x, y + \Delta y) \quad (16.66)$$

$$[\Delta x, \Delta y]^T = \arg \min_{(\Delta x, \Delta y)} MPC(\Delta x, \Delta y) \quad (16.67)$$

运动估计值 $\mathbf{r} = (\Delta x, \Delta y)$ 对应匹配像素的最大数量 MPC 准则需要一个阈值比较器和 $\log_2(m \times n)$ 计数器 [Gha90]。

(2) 搜索策略

为了求得最佳位移估计，可以计算所有可能的位移矢量对应的匹配误差，然后选择最小匹配误差对应的矢量就是最佳位移估计值，这就是全搜索策略。这种策略的最大优点是找到全局最优值，但十分浪费时间，因此，人们提出了各种快速搜索策略。尽管快速搜索策略得到的可能是局部最优值，但由于其快速计算的实用性，在实际中得到了广泛的应用。下面讨论一种快速搜索方法： n 步搜索或对数搜索。

设窗口大小为 15×15 ，当前像素值位于窗口中心，用 "0" 来标记，如图 14.18(a)所示。第一步，选择标记为 "0" 和 "1" 的 9 个像素计算匹配准则函数，如果最佳匹配仍在 "0" 处，则无运动。第二步，以第一步最佳匹配对应的像素点为中心选择 8 个点（图中用标记 "2" 表示），计算这 8 个点的匹配准则函数值。第三步，以第二步最佳匹配对应的像素点为中心选择 8 个点（图中用标记 "8" 表示），计算这 8 个点的匹配准则函数值，最佳匹配值即为最后的最佳运动估计。由图 14.19(a)可见，每进行一步，搜索距离减小一半，并且愈来愈接近精确解。人们将上述搜索过程称为 3 步搜索。当然可以继续在于像素级上进行搜索，以得到更精确的估计值，这样就需要大于 3 步的搜索，称之为 n 步搜索，由于搜索步数与窗口内像素个数是对数关系，因此，常将这种搜索称为对数搜索。另一种对数搜索策略是在每一步有 4 个搜索位置，它们以十字形或交叉形布置，如图 16.19(b)所示。

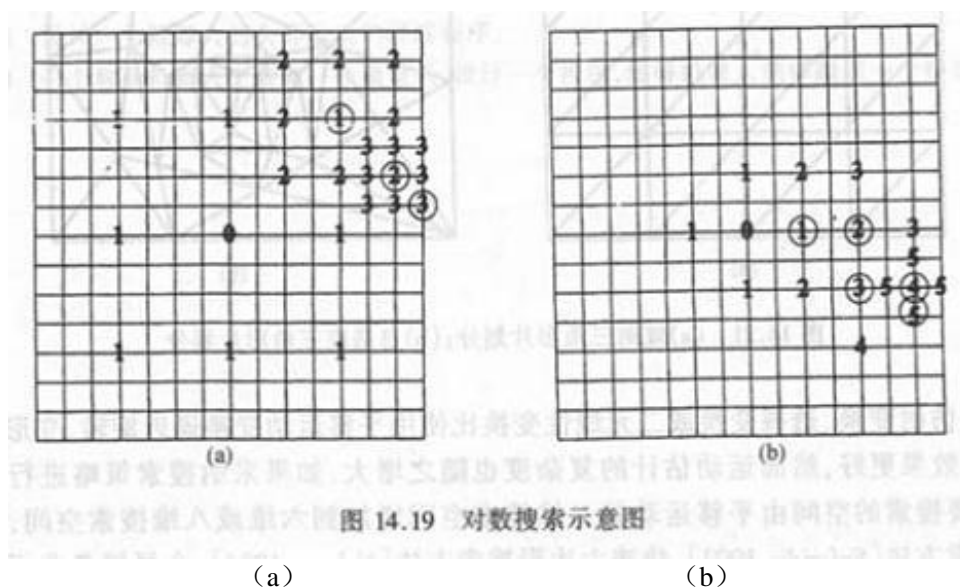


图 14.19 对数搜索示意图

*14.4.5 广义图像运动估计

虽然基于平移的块模型的运动估计是简单的，但处理旋转、变形运动和不连续运动场时，效果不是很好。下面介绍处理一般运动的广义图像运动分析。

在广义块运动匹配中，首先将当前帧分解成三角片、矩形片和任意四边形片。然后在给定的空间变换下，在另一帧中找到最佳匹配的三角片或四边形片，如图 14.20 所示。空间变换与分解片的形状有关，三角片提供了足够的自由度使用仿射变换，因为每一个顶点提供两个方程，而仿射变换有六个参数。透视变换和二次线性变换包含有八个参数，非常适合用四边形片。如果对四边形作仿射变换，将产生超定方程，即平行线经变换后仍然是平行线。

使用仿射变换、透视变换或二次线性变换比使用平移运动变换逼近旋转、变形以及缩放运动的效果更好，然而运动估计的复杂度也随之增大。如果采纳搜索策略进行运动估计，则需要搜索的空间由平移运动的二维搜索空间增加到六维或八维搜索空间。全搜索方法[sef93]，快速六边形搜索方法[Nak94]，全局样条曲面拟合方法[Flu92]。全搜索方法可以归纳为如下几步：

- I. 将图像分割成矩形片（三角形片）
- II. 从图像的初始估计值开始，摄动匹配矩形片（三角片）的顶点坐标
- III. 对于每一个矩形片（三角形片），求预定的空间变换参数，使得通过四个（三个）匹配顶点将上一帧的矩形片（三角形片）映射成当前帧的矩形片（三角形片）。
- IV. 使用空间变换求矩形片（三角形片）内对应象素的坐标，然后求解待匹配片之间的 MSE 或 MAD 准则函数。
- V. 选择 MSE 或 MAD 最小值对应的空间变化。

上面的算法实际上隐含了这样一个假设：矩形片（三角形片）内的象素应属于同一个运动的物体，否则，无法用一个运动模型来描述图像运动。对于图像中包含有多个运动物体时，应该使用自适应矩形片或三角形片，如图 14.21。

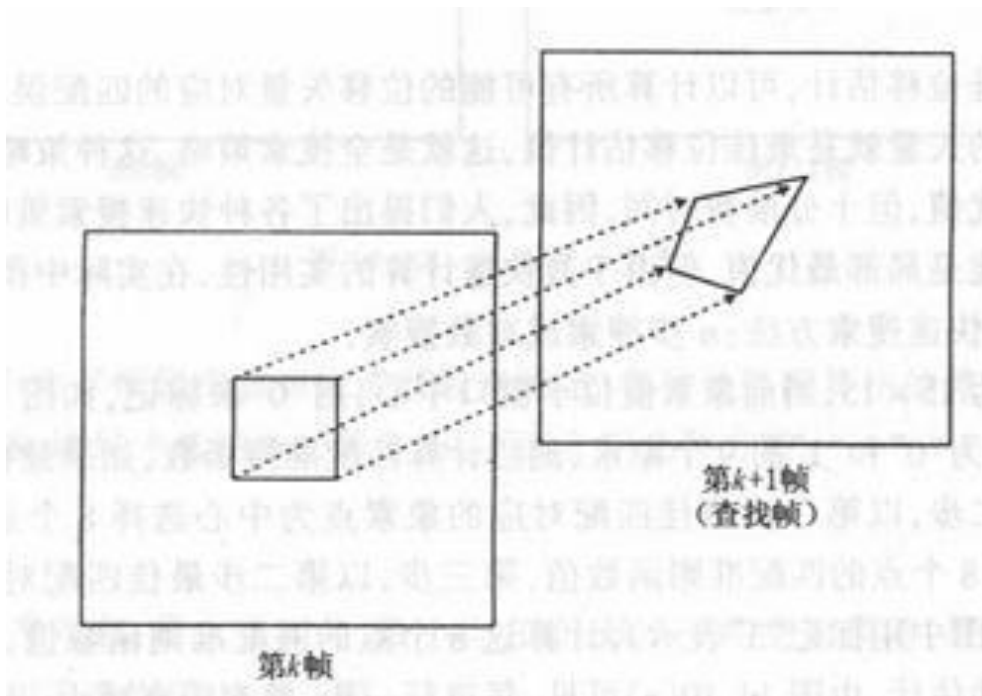


图 14.20 广义运动匹配示意图

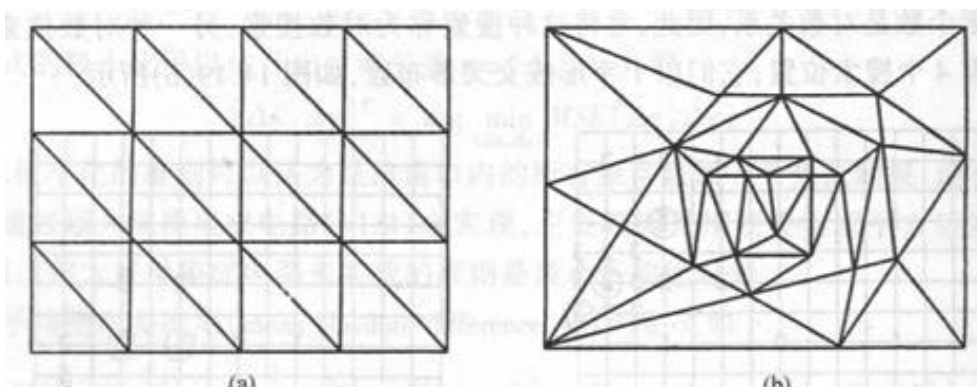


图 14.21 (a) 规则三角形片划分, (b) 自适应三角形片划分

思考题

- 14.1 什么是累积差分图像？它与差分图像有何不同？
- 14.2 时变边缘和运动边缘的不同之处是什么？在动态场景中你如何检测运动边缘？
- 14.3 为得到亚像素分辨率的特征点位置，计算从 Moravet 兴趣算子第 2 步中得到的最小变化的中继图像中计算 x 和 y 要素。
- 14.4 什么是光流？它与速度场有何区别？
- 14.5 请讨论孔径问题，(1) 单一像素匹配，(2) 线条匹配，(3) 曲线匹配，(4) 角点匹配。
- 14.6 一个 6×8 像素的左上角最初置于 $(0, 0)$ ，在以匀速经过四图像后，它运动至 $(4, 8)$ ，找出绝对、累积、正、负差分图像。
- 14.7 比较全搜索和 n 步搜索的复杂度。

计算机练习题

- 14.1 布置一个场景，场景内有两个朋友相互走近，并握手寒暄。获取该场景的一个灰度图像序列，并设计一个基于图像累积差分方法的程序提取图像序列的目标。
- 14.2 设计一个求解对应点的程序，其中图像中至少包含三个物体，并且物体以不同的方向运动。
- 14.3 设计一个跟踪人的头部运动的跟踪程序。
- 14.4 一个移动机器人上装有一个摄像机，设计一个程序，使得机器人能够跟踪一个特定的运动物体。