

第 3 章

基于手势识别的手势跟踪

手势的识别就是根据用户的手势识别手势的含义。基于视觉的手势识别以人手直接作为计算机的输入设备,人机之间的通信将不再需要中间媒体,用户只需定义适当的手势来对周围的机器进行控制。但是,由于手势本身具有多样性、多义性以及时间和空间上的差异性等特点,加之人手是复杂变形体以及视觉本身的不适定性,基于视觉的手势识别是一个富有挑战性的、多学科交叉的研究课题,也是完成手势跟踪任务的重要途径。

早在 1991 年,富士通实验室就完成了对 46 个手势符号的识别工作^[142]。J. Oavis 和 M. Shah 将戴上指尖具有高亮标记的视觉手套的手势作为系统的输入,可识别 7 种手势^[143]。Starner 等在对美国手势中带有词性的 40 个词汇随机组成的短句子进行识别,识别率达到 99.2%^[144]。K. Grobel 和 M. Assam 从视频录像中提取特征,采用 HMM 技术识别 262 个孤立词,正确率为 91.3%^[145]。

我国的高文等给出了 ANN 与 HMM 的混合方法作为手势的训练识别方法^[146],将 ANN-HMM 混合方法应用于有 18 个传感器的 CyberGlove 型号数据手套的中国手势识别系统中,孤立词识别率为 90%,简单语句级识别率为 92%^[147]。接下来他们又选取 Cyberglove 型号数据手套作为手势输入设备,并采用了 DGMM (Dynamic Gaussian Mixture Model) 作为系统的识别技术,即利用一个随时间变化的具有 M 个分量的混合 Gaussian-N 元混合密度来模型化手势信号,可识别中国手势字典中 274 个词条,识别率为 98.2%^[148]。

清华大学的祝远新、徐光佑等给出了一种基于视觉的动态孤立手势识别技术,借助于图像运动的变阶参数模型和鲁棒回归分析,提出一种基于运动分割的图像运动估计方法。他们给出了有关连续动态手势的识别,融合手势运动信息和皮肤颜色信息,进行复杂背景下的手势分割。通过结合手势的时序信息、运动表现及形状表现,提出动态手势的时空表现模型,并提出基于颜色、运行以及形状等多模式信息的分层融合策略抽取时空表现模型的参数。最后提出动态时空规整算法用于手势识别,对 12 种手势,平均识别率高达 97%^[69]。

文献[149]使用概率神经网络(PNN)为静态手势分类建立了一种基于非线性降维方法的手势识别与跟踪视觉系统。文献[150]描述了基于 3D 数据的手势识别系统,该系统采用一种可以产生密集图像序列的 3D 传感器获取数据。文献[151]K 近邻分类技术实现基于在线视频的手写阿拉伯字母识别方法,该方法按照二维离散余弦变换、区域带状编码换和低通滤波,得到的特征向量是独立于时间的,识别率达到 99%。文献[152]采用由粗到精的方法,通过滑动窗技术实时提取手势的统计特征,实现手势类别的粗略划分,然后采用不同方法对各类手势进行分析。

3.1 多灰度图像连续形变的计算机识别技术研究^[153]

3.1.1 基本背景

国外进行了基于相位转移技术(phase-shifting techniques)的连续形变识别算法在度量衡领域的应用研究^[154], Colonna 对其要求相位须保持稳定性的缺点进行了改进,其主要手段是取相继的5帧图像中对应点的相位共同决定该点的形变强度^[155]。但逐点进行处理限制了其处理速度,也必然影响到相邻两帧之间的时间间隔,同时其形变模型具有很大的局限性以及相邻两帧之间存在线性关系之假设的不合理性。朱仲涛等人提出了一种通过研究图像处理算子而对图像进行处理的新方法,其结果是得到了边缘算子的微分不变量或准不变量^[156,157],但取得形变的变换群不是一件容易的事,还有一定的理论难度。从总体上看,大量的研究成果集中在各种类型的不变量的提出和应用,它们成为处理形变图像的主要方法^[146-148]。但充分条件与必要条件之间的矛盾、特殊性与普适性之间的矛盾以及算法的精度与误差度之间的矛盾等难以协调。文献[160]中提出了能量最小化方法进行形变图像的匹配,通过把形变的雅可比(Jacobian)的值限制在一定的范围内而保证其同胚性(homeomorphism),虽然有关计算只在图像的边界上进行,但边界信息的获取仍要占相当的时间,只考虑到了算法的准确性,而没有注意到在实践中往往可以用一定的粗糙性(即允许一定的误差)换取时间。

已经有人把网格作为一种工具和手段引入到连续形变的研究中^[161],但对于网格和连续形变之间的关系还研究得不够深入。

为此,我们提出了一种新的相似不变量,在对物体概念加以限定的条件下用相似不变量进行物体连续形变的识别问题进行了探讨,但对物体概念上的限定必然限制其使用范围,二是得到物体的“圈体”,其时间复杂度难以改善,故有进一步研究的必要^[162]。

连续形变图像的基本内涵是在变化前后,图像在空间上和时间上须同时保持连续,所以其识别的结果与算法本身效能与性能和采样设备的速度与质量等有关。显然,对存在误差的采样数据进行精确的处理是没有必要的。另外,在实践中没有必要也不可能满足理论上的高精度。可见,存在改善算法时间复杂度的可能性和必要性,从而有可能进一步缩短相邻两帧之间的时间间隔,从另一种更好的角度提高识别的准确性和可靠性,本节将采用不断细化的网格化技术来达到改善算法的时间复杂度的目的。

又由于客观世界的复杂性和多样性,有的形变自始至终保持时间和空间上的连续性,例如由物理属性相似的部件所构成物体的膨胀或收缩过程,大多数属于这种情形。有的形变在连续形变过程中在局部范围内又进行非连续形变,例如人脸逐渐微笑的过程就属于此种情形。通过引入拓扑保持性连续形变和拓扑非保持性连续形变把这些对立的矛盾统一起来,从而满足不同层次、不同场合的应用需要。

3.1.2 基本术语

1. 压线格

用 $M \times N (M > 0, N > 0)$ 条互相垂直的两组直线对图像进行逻辑上的分割,图像被分割

成 $(M+1) \times (N+1)$ 个小矩形,每个矩形被称为一个网格。沿着网格的4条边按一定顺序扫描图像,沿途所经历不同的灰度值序列、不同灰度值的数目以及所感兴趣的其他信息等构成该网格的网格属性,每个网格左上角的坐标称为该网格的坐标(简称坐标)。把网格属性中不同灰度值的数目大于等于1的网格称为压线格。图3.1中的网格A就是图像的一个压线格。图3.2是一个压线格程序运行实例。

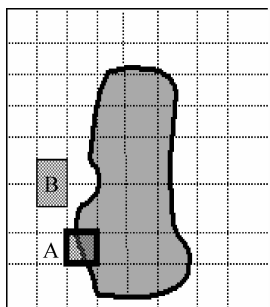


图 3.1 压线格示意图

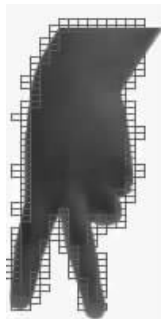


图 3.2 一个压线格实例

2. 网格图像的顶点网格和网格图像的边图

若无论怎样细分某网格,总可以得到一个且仅有一个属性相同的灰度数目等于2的子网格,则该网格被称为该网格图像的一个边网格。若无论怎样细分某网格,总可以得到一个且仅有一个属性相同的灰度数目大于等于3的子网格,则该网格被称为该网格图像的一个顶点网格。第 I 帧以点 s 为邻域中心的顶点网格记作 $h_{i,s}$ 。若对很小尺寸的某网格的灰度数目等于2,则该网格被称为该网格图像的一个边网格,由相邻且灰度值的数目为2、灰度值序列相同的压线格所构成的网格图像的子图称为该网格图像的一个边图。显然,任意一个网格的边图的两端一定是顶点网格,以 $h_{i,s1}$ 和 $h_{i,s2}$ 为端点的边图记作 $\langle h_{i,s1}, h_{i,s2} \rangle$ 。另外,所有顶点网格和边图构成网格图像。图3.3中,用矩形框圈起来的网格图就是网格图像的顶点,剩下圈起来的部分就是网格图像的边图。



图 3.3 顶点网格与边图示意图

3. G_i 与 G_{i+1} 同胚

如果 G_i 和 G_{i+1} 之间满足同构^[163]和有序相邻这两个条件,则称 G_i 与 G_{i+1} 同胚。所谓有序相邻,是指具有相同顶点的两条不同的边在形变前后的相对位置关系保持不变的性质。

从这一定义出发, G_i 与 G_{i+1} 同胚的充要条件是:

- (1) 它们的顶点网格和边图之间分别具有一一对应的关系。
- (2) 有序相邻。

值得注意的是, G_i 与 G_{i+1} 的边图之间具有一一对应的关系,并不意味着边图的各网格之间必须存在一一对应的关系,事实上,对应网格的数量不一定相等。图3.4和图3.5

所示的两个网格图像不同胚,因为不满足有序相邻性条件。

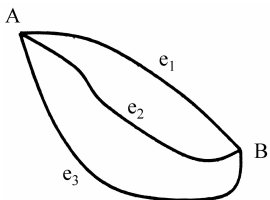


图 3.4 一个假想的顶点网格和边图

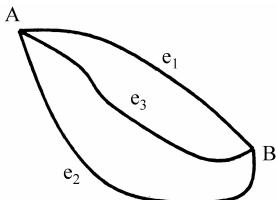


图 3.5 另一个假想的顶点网格和边图

4. 连续形变

设第 i 帧图像为 I_i , 第 $i+1$ 帧图像为 I_{i+1} , 两帧之间的时间间隔为 Δt , 若满足:

$$\lim_{\Delta t \rightarrow 0} I_{i+1} = I_i \quad (3.1)$$

且 G_i 与 G_{i+1} 同胚, 则称图像由第 i 帧到第 $i+1$ 帧是一次连续形变。如果设当前帧的图像边界向量为 C_{current} , 上一帧的图像边界向量为 C_{last} , 则式(3.1)等价于: 对任意小的正数 ϵ , 都有以下包含关系成立。

$$C_{\text{in}} \subseteq C_{\text{current}} \subseteq C_{\text{out}} \quad (3.2)$$

其中, C_{in} 、 C_{out} 分别表示 C_{in} 和 C_{out} 所表示边界的点集, $C_{\text{in}} = C_{\text{last}} - \epsilon N$, $C_{\text{out}} = C_{\text{last}} + \epsilon N$, N 为 C_{last} 上点的法线方向。图 3.6 中表示了 C_{in} 、 C_{out} 和 C_{last} 三者之间的关系。以下也称 C_{current} 夹在以 ϵ 为厚度的条状区域内。当式(3.1)中的 $\Delta t \rightarrow 0$ 时, 式(3.2)中的包含关系一定成立; 反之, 对任意小的正数 ϵ , 有式(3.2)成立, 则 $C_{\text{current}} \rightarrow C_{\text{last}}$, 即式(3.1)成立。

若给定的形变图像在给定的时间范围内任意两相邻帧之间都是连续形变的, 谓之跟踪成功; 否则, 跟踪失败。

5. 网格属性

网格所覆盖图像的灰度序列、灰度数目、网格的长度和网格的高度等统称为网格属性。网格左上角的坐标称为网格坐标。图 3.7 中, 以由字母 A 所表示的网格为例, $((0, 80), 2, (20, 20))$ 就是该网格的网格属性。其中 $(0, 80)$ 表示网格所覆盖的两个灰度值, 2 表示网格所覆盖的灰度数目, $(50, 20)$ 表示该网格左上角的坐标, $(20, 20)$ 表示网格的长度和高度。

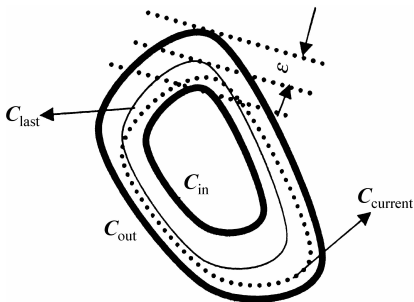


图 3.6 连续形变的概念

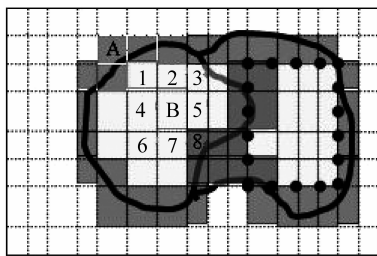


图 3.7 由不同灰度区域构成的假想图像

6. 条件 NS(No Separation)

对第 i 帧 F_i 的任意压线格 (i, s, a) (s 为该压线格的坐标, a 为该网格的属性), $U(i, s)$ 为压线格 (i, s) 的八邻域, 如果

$$U(i+1, s) \supset (i, s, a) \quad (3.3)$$

则称压线格 (i, s) 满足条件 NS。图 3.7 中的网格 1, 2, ..., 8 即为网格 B 的八邻域。

7. 条件 NC(No Conglutination)

对第 $i+1$ 帧 F_{i+1} 的任意压线格 $(i+1, s, a)$ (s 为该压线格的坐标, a 为该网格的属性), 如果

$$U(i, s) \supset (i+1, s, a) \quad (3.4)$$

则称压线格 (i, s) 满足条件 NC。

8. 免检网格

在允许的精度下, 无需再进行细分的网格称为免检网格。图 3.7 中的字母 B 所在的网格就是免检网格。免检网格的提出可以避免不必要的进一步分解和计算。

9. 区域的内接包围盒

包围连续免检网格的、其长度和高度方向分别与坐标轴平行的、长宽比最接近于网格的长宽比的最大矩形称为该区域的内接包围盒。图 3.7 中, 由黑点所形成的矩形就是右半部分区域的内接包围盒。

10. 拓扑保持性连续形变

如果在图像的连续形变过程中, 由图像的灰度所决定的各区域之间的拓扑连接关系保持不变, 称为拓扑保持性连续形变。由图 3.8 直接变到图 3.10 (不经过图 3.3) 即是拓扑保持性连续形变。所谓拓扑连接关系, 指各区域本身的拓扑结构的同伦 (Homotopic) 以及构成图像的各区域之间的邻接关系。

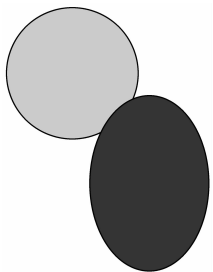


图 3.8 由三个灰度区域构成的图像

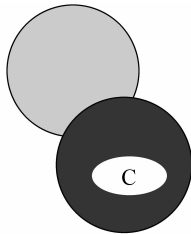


图 3.9 图 3.8 的一次可能形变

11. 拓扑突变

指各区域本身的拓扑结构的同伦 (Homotopic) 性或构成图像的各区域之间的邻接关系

遭到破坏。典型的情形是区域内部或区域之间发生了分离 (Separation) 或粘连 (Conglutination) 致使拓扑突变的发生。图 3.12 是图 3.11 的一次可能的拓扑突变。

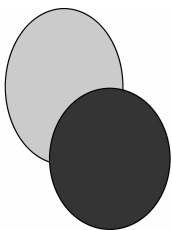


图 3.10 图 3.8 的另一次可能形变

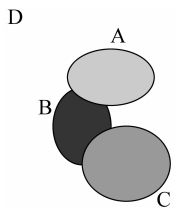


图 3.11 4 区域图像

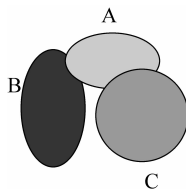


图 3.12 图 3.11 的一次可能的拓扑突变

12. 拓扑非保持性连续形变

允许图像在一定范围、一定时间内的拓扑突变,即在此范围和程度下形变过程不受拓扑保持性连续形变的制约,但除此之外图像的形变须是拓扑保持性连续形变。由图 3.8 变到图 3.10,若 C 在规定的范围和时间内,则就是拓扑非保持性形变。

3.1.3 识别算法

算法的总体流程如图 3.13 所示。

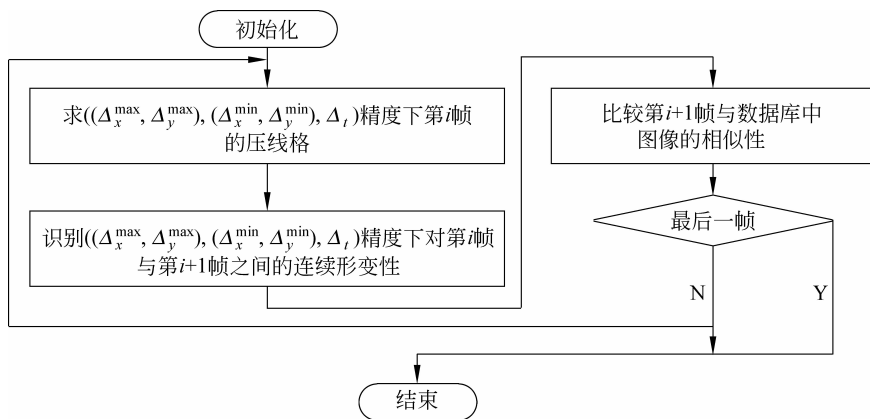


图 3.13 算法的总体流程图

3.1.4 基本定理

定理 相邻两帧是拓扑保持性连续形变的充要条件是: 同时满足条件 NS 和条件 NC。

证明: 先证充分性。若满足条件 NS, 则形变前后无分裂发生, 即形变之前是相连的拓扑关系在形变之后仍然保持其连接关系; 否则, 若形变后有关系被断裂, 则在第 $i+1$ 帧的断点附近不存在与形变前具有相同属性的网格, 这与条件 NS 的定义矛盾。另一方面, 若条件 NC 满足, 则形变后无粘连发生, 即无新的连接关系发生; 否则, 若增加了新的连接关系, 则在第 i 帧新增关系附近存在一个邻域, 不包含具有第 $i+1$ 帧同一位置网格属性的任何网

格,这与 NC 的定义矛盾。

所以,根据拓扑保持性连续形变的定义,充分性得到证明。

再证必要性。事实上,若相邻两帧是拓扑保持性连续形变,则必然拓扑同胚,从而形变前后拓扑关系不分裂、不粘连,又从充分性的证明看到,不分裂、不粘连与条件 NS 和条件 NC 是同义语,所以必要性正确。

基于上述定理,可以对算法进行进一步的描述。

3.1.5 压线格算法

求压线格算法的主要目标就是确定在给定精度下的压线格集合,其实质是得到已知图像的具有一定误差度的边界,可以看到,这种误差可以根据应用需要加以调节。求压线格的具体算法描述如下:

(1) 清空新压线格集合。

(2) $\Delta_x \leftarrow \Delta_x/2, \Delta_y \leftarrow \Delta_y/2$, 若满足条件 $(\Delta_x, \Delta_y) > (\Delta_x^{\min}, \Delta_y^{\min})$, 转第(3)步; 否则, 转第(9)步。

(3) 把当前压线格中的每个网格一分为 4, 其结果是每个子网格的长度和高度分别为 Δ_x 和 Δ_y 。

(4) 求每个子网格的网格属性。

(5) 若任意网格 s 的网格属性中的灰度数不为 1, 转第(7)步; 否则, 转第(6)步。

(6) 是否满足条件 $(\Delta_x, \Delta_y) > (\Delta_x^{\max}, \Delta_y^{\max})$? 若是, 转第(8)步; 否则, 转第(2)步。

(7) 将该网格加入新压线格集合中, 转第(2)步。

(8) 返回。

作为初始条件, Δ_x 为原图像的宽度, Δ_y 为原图像的高度, Counter=0。

3.1.6 连续形变识别

把连续形变分为拓扑保持性连续形变和拓扑非保持性连续形变, 首先对这两种情形分别加以讨论。

1. $((\Delta_x^{\max}, \Delta_y^{\max}), (\Delta_x^{\min}, \Delta_y^{\min}), \Delta_t)$ 精度下的拓扑保持性连续形变

一般情况下, 由计算机根据连续形变的定义进行绝对精确的判断是不可能的, 因为至少相邻的帧之间总有一定的时间间隔 Δ_t , 所以只能进行 $((\Delta_x^{\max}, \Delta_y^{\max}), (\Delta_x^{\min}, \Delta_y^{\min}), \Delta_t)$ 精度下的拓扑保持性连续形变识别。拓扑保持性连续形变在客观上要求压线格在形变过程中不得发生分离(Separate)和粘连(Conglutination), 即要求在时间间隔为 Δ_t 的两相邻帧之间对条件 NS 和条件 NC 进行检验, 具体的算法描述为:

(1) 初始化当前网格 s 。

(2) 是否满足条件 NS? 若满足, 转第(3)步; 否则, 转第(6)步。

(3) 是否满足条件 NC? 若满足, 转第(4)步; 否则, 转第(6)步。

(4) $s \leftarrow$ 下一格。

(5) 是否所有压线格都已处理完? 若是, 转第(7)步; 否则, 转第(2)步。

(6) 识别失败, 返回。

(7) 识别成功, 返回。

条件 NC 客观上要求在形变过程中, 边界不得发生粘连或原来不相交的边界变成相交。

图 3.14~图 3.17 就是条件 NC 不满足的情形。



图 3.14 第 I 帧图像

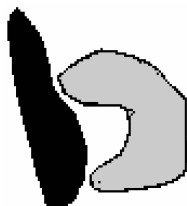


图 3.15 第 I 帧可能的形变结果



图 3.16 第 I 帧可能的另一种形变结果



图 3.17 第 I 帧图像的一种可能形变结果

2. $((\Delta_x^{\max}, \Delta_y^{\max}), (\Delta_x^{\min}, \Delta_y^{\min}), \Delta_t, \Sigma)$ 精度下的非拓扑保持性连续形变的处理

非拓扑保持性连续形变的基本内涵是在拓扑保持性连续形变过程中允许局部范围内的拓扑突变。根据拓扑保持性连续形变的算法, 很容易得到 $((\Delta_x^{\max}, \Delta_y^{\max}), (\Delta_x^{\min}, \Delta_y^{\min}), \Delta_t, \Sigma)$ 精度下的非拓扑保持性连续形变识别算法。

- (1) 初始化当前网格 s 。
- (2) 是否满足条件 NS? 若满足, 转第(3)步; 否则, 转第(6)步。
- (3) 是否满足条件 NC? 若满足, 转第(4)步; 否则, 转第(6)步。
- (4) $s \leftarrow$ 下一格。
- (5) 是否所有压线格都已处理完? 若是, 转第(7)步; 否则, 转第(2)步。
- (6) $T \leftarrow T + 1$ 。
- (7) 如果 $\frac{T}{\text{TOTAL}} < \Sigma$, 则识别成功, 返回; 否则, 识别失败, 返回。

算法一旦发现了拓扑突变, 首先统计突变网格的个数 T , 当所有的帧都处理完毕后, 只要 T 与网格总数 TOTAL 之比在一定的范围内, 则认为发生了在 $((\Delta_x^{\max}, \Delta_y^{\max}), (\Delta_x^{\min}, \Delta_y^{\min}), \Delta_t, \Sigma)$ 精度下的非拓扑保持性连续形变, 识别成功。从实际情况来看, 把突变的范围加以一定的限制是可行的, 也是必要的, 因为这样可以满足大多数情况的应用需要。图 3.17 就是图 3.16 的一种可能的非拓扑保持性连续形变。

3.1.7 与数据库中已知图像的匹配

事先把一些典型的静态图像存于数据库中, 在对每一幅图像进行连续形变或非连续形变的处理后, 都将其与数据库中的图像进行 NS、NC 条件的判定, 如果在连续的图像序列

中,只要有一个图像与数据库中的一个图像能够匹配,则继续判断最后一帧与第一帧是否匹配,若是,则识别成功。图 3.18 给出了识别成功的时间状态图,从而把不同的物体区别开来。

3.1.8 算法性能分析及实验结果

假设压线格的个数是 $M \times N$,则需要经过 $8 \times [M(N+1) + N(M+1)]$ 次比较才能得到所有压线格的属性,从而获得压线格。进行一次条件 NC 检测需要比较 $9d_i, d_i$ 表示第 $i+1$ 帧网格 t 的不同灰度值,一般情况下为 2;进行一次条件 NS 检测需要比较 $9n_i, n_i$ 表示第 i 帧网格 t 的不同灰度值。一般情况下, d_i 或 n_i 的值为 2。故进行拓扑保持性连续形变算法识别需要进行比较的总次数是

$$\begin{aligned} T_G &= 8[M(N+1) + N(M+1)] + 9MN(D_i + N_i) \\ &= MN(16 + 9D_i + 9N_i) + 8(M+N) \end{aligned} \quad (3.5)$$

其中, D_i 和 N_i 分别是第 $i+1$ 帧和第 i 帧所有网格灰度的最大值。显然

$$T_G \leq \frac{(M+N)^2}{4}(16 + 9D_i + 9N_i) + 8(M+N) \quad (3.6)$$

可见,比较次数的上确界主要受 $M+N$ 的影响, $(16 + 9D_i + 9N_i)$ 基本上是常数。设每个网格的宽度和高度分别为 w 和 h ,则按照传统的逐点比较算法^[142],比较的总次数为 $T_P = whMN$,注意到

$$\frac{T_G}{T_P} \leq \frac{16 + 18T}{wh} + \frac{8(M+N)}{whMN} \quad (3.7)$$

其中, T 为 D_i 和 N_i 的最大值。因为 wh 是以像素为单位的一个网格的面积,当 M 和 N 小到一定程度时,有

$$\frac{16 + 18T}{wh} \rightarrow 0 \quad (3.8)$$

于是

$$\frac{T_G}{T_P} \approx \frac{8(M+N)}{S} \quad (3.9)$$

S 为原图的面积。显然,当 M 和 N 小到一定程度时,总有 $T_G \leq T_P$ 。而且,当划分越少时,提出算法的效率越高。另外,非拓扑保持性连续形变算法的时间分析与上面有相似的结果。精度 $((\Delta_x^{\max}, \Delta_y^{\max}), (\Delta_x^{\min}, \Delta_y^{\min}))$ 实际上由 M 和 N 控制。

另外,进一步将本节提出的算法与文献[158]所给算法进行比较。在文献[158]中,每次识别的时间复杂度为 $O(s \times n)$,其中 s 为物体中圈的数量, n 为每个圈的顶点个数。其中的 $s \times n$ 表示源图中顶点的总数,在一定的条件下,时间复杂度可以得到有效改善。另外,原算法与源图的拓扑结构有关,而这里的 $(M+N)^2$ 主要与网格的规模有关,是本算法的一个重要特点。

对图 3.19 中的连续形变序列分别用文献[158]中提出的方法与本节方法进行比较,采用本节算法每比较相邻两帧的比较次数是 5300(压线格取为 100 个),采用文献[162]中算法比较相邻两帧的比较次数是 4800。但如果图像的拓扑结构复杂度增加,采用本节算法的比较次数不变,采用文献[162]中提出的算法的比较次数将增加。

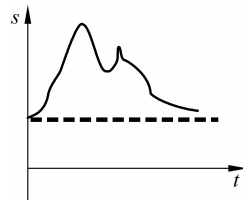


图 3.18 识别成功的时间状态图

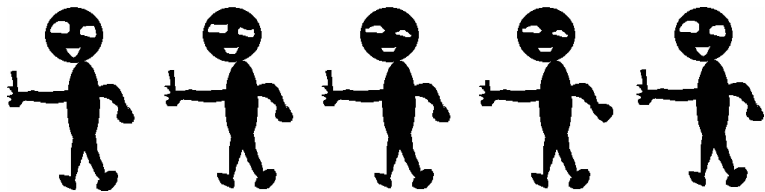


图 3.19 一个连续形变序列

一方面考虑到客观世界的复杂性、丰富性和多样性,另一方面注意到样本采集过程中的误差,以及计算机本身处理速度的限制,所以要找到一种具有普适性的识别二维连续形变图像识别算法是不容易的。快速提高算法的处理速度,具有一定普适性并能基本上满足不同层次、不同精度的需要是处理连续形变领域的重要问题,本节旨在就此问题从新的角度作一些探讨。

提出的算法的主要特点在于:

(1) 与现有的有关算法相比,本节提出的算法的时间复杂度得到极大的改善,可以看出,算法的时间复杂度不主要依赖于图像拓扑结构的复杂度。事实上,设图像的高度和宽度分别为 H 和 W ,则获得边界信息所需最长时间为 $\log_2(HW)$,即与图像拓扑结构的复杂性没有关系,而用其他方法,如边界跟踪法,其时间复杂度显然与图像拓扑结构的复杂性有关。其次,在拓扑结构比较部分,其时间复杂度也只与压线格的个数有关。

(2) 根据不同的应用层次和不同的应用需要,算法的精度可以很灵活地调节,既可以精确到一个像素,又可以粗糙到任意大小的矩形块,从而使得满足不同的应用场合的需要成为可能。

(3) 既可以处理拓扑保持性连续形变,又可以处理非保持性连续形变,从而把对不同类型的连续形变的处理较好地统一起来。

(4) 形变图像的边界既可以封闭,也可以是一段不封闭的曲线或直线;形变既可以是线性的,也可以是非线性的,即对任意复杂的拓扑结构,算法都是有效的。

我们看到,识别成功的条件是以下几个条件同时成立:

- (1) 任意两个相邻帧之间是连续形变;
- (2) 在形变过程中至少有一帧与已知数据库中的某图像相匹配;
- (3) 最后一帧与第一帧可以相匹配。

很显然,条件(1)保证了拓扑结构的不变性以及时间上的连续性,条件(2)保证了识别结果的唯一性,条件(3)把不同的物体区别开来。这样,把形变的起点、终点和过程统一起来;把静态的数据库与动态的形变统一起来。

3.2 基于连续形变理论和方法的手势识别技术

连续形变理论和方法是处理形变手势识别的基本方法之一。在基于自然手的新型 HCI 理论和应用中,不可能要求用户每时每刻都要小心翼翼地保持某种僵硬的手势,以免计算机不识别或误识别。所以,探讨具有“柔性”的手势识别方法,使得用户的手势可以在一定的幅度和范围内灵活变化具有重要意义。

用 $W(I, y, R)$ 表示第 I 帧中以网格 y 为中心、以 R 为尺寸的邻域内的、相邻网格的集合。若网格 x 在 $W(I, y, R)$ 内, 则称 $x \subseteq W(I, y, R)$ 。我们约定, 当且仅当某网格的坐标落在某邻域内时, 称该网格在该邻域内。

3.2.1 识别算法^[164]

首先给出以下性质:

性质 1 网格图像细分的极限就是原图像的边界图, 如果原图像的边界存在的话。

证明: 设 $G(k+1)$ 是 $G(k)$ 的一个细分, 则

$$G(1) \supset G(2) \supset G(3) \supset \cdots \supset G(s) \supset \cdots \quad (3.10)$$

显然, 如果原图像的边界图存在的话, 则随着 s 的无限增大, 上述序列必收敛, 其极限就是原图的边界图。

性质 2 I_{i+1} 是 I_i 的一次连续形变的必要条件是: G_i 中所有网格的属性的集合 $= G_{i+1}$ 中所有网格的属性的集合。

证明: 设 I_{i+1} 是 I_i 的一次连续形变。据连续形变的定义, G_i 与 G_{i+1} 必同胚, 从而 G_i 与 G_{i+1} 同构, 顶点网格之间可以建立一一对应的关系, 边图之间也有一一对应的关系, 而这种一一对应都是基于网格属性的, 所以结论成立。

性质 1 从理论上说明了用网格的手段来代替原图像边界的可行性以及在许可精度下的合理性。为了进一步揭示连续形变图像所对应的网格图像之间的内在联系, 首先给出定理 1。

定理 1 I_{i+1} 是 I_i 的一次连续形变的充要条件是以下两个条件同时得到满足:

① 存在任意小的 $r_1 > 0$, 使得对 $\forall x \in G_i$, 存在 $y \in G_{i+1}$, 使得 $x \in W(I+1, y, r_1)$;

② 存在任意小的 $r_2 > 0$, 使得对 $\forall y \in G_{i+1}$, 存在 $x \in G_i$, 使得 $y \in W(I, x, r_2)$ 。

其中, x 和 y 的网格属性相同。

证明:

(1) 充分性的证明

首先注意到, 若所述两个条件同时满足, 则 G_i 与 G_{i+1} 必同胚。根据条件(1), $\forall x \in G_i$, 存在 $y \in G_{i+1}$, 使得 $(I, p_x, B) \subseteq W(I+1, y, r_1)$, p_x 是 x 的网格坐标, 所以 G_i 中属性的集合 $\subseteq G_{i+1}$ 中属性的集合; 根据条件(2), $\forall y \in G_{i+1}$, 存在 $x \in G_i$, 使得 $(I+1, p_y, B) \subseteq W(I, x, r_2)$, p_y 是 y 的网格坐标, 所以 G_i 中属性的集合 $\supseteq G_{i+1}$ 中属性的集合。于是 G_i 中所有网格的属性的集合 $= G_{i+1}$ 中所有网格的属性的集合, 从而网格图像 G_i 与 G_{i+1} 的顶点网格和边图之间可以建立一一对应的关系。另一方面, r_1 和 r_2 可以取任意小, 当然可以远远小于不同的顶点网格和不同的边图之间的距离, 在题设中的两个包含关系同时成立的条件下, 必然保持有序相邻, 否则所述的“两个包含关系同时成立”与“ r_1 和 r_2 任意小”之间将发生矛盾。

另外, 若取 $r = \max(r_1, r_2)$, 则 G_i 和 G_{i+1} 都将夹在以 r 为厚度的条状区域内, 即 $\lim_{\Delta r \rightarrow 0} I_{i+1} = I_i$ 成立。

(2) 必要性的证明

设 G_i 与 G_{i+1} 同胚且 $\lim_{\Delta r \rightarrow 0} I_{i+1} = I_i$ 。首先, G_i 和 G_{i+1} 的顶点网格之间可以建立一一对应的关系, 所以, 第一, 若 x 是 G_i 的任意一个顶点网格, 则必存在 G_{i+1} 中与 x 对应的顶点网格是

y , 并且 x 与 y 具有相同的网格属性, 考虑到对任意小的 $\epsilon > 0$, G_i 和 G_{i+1} 都将夹在以 ϵ 为厚度的条状区域内, 所以 $|x-y| < \epsilon$, $x \subseteq W(I+1, y, \epsilon)$, 这就证明了定理中的条件①; 同理, 可证条件②成立。第二, 若 x 是 G_i 中的一个边图 $\langle h_{i,s1}, h_{i,s2} \rangle$ 上的任意一个网格, 则在 G_{i+1} 中存在一个边图 $\langle h_{i+1,s1}, h_{i+1,s2} \rangle$ 与之对应。若存在 $r > 0$ 和任意 $y \in G_{i+1}$, 都有 $x \notin W(I+1, y, r)$, 则 $|x-y| > \epsilon$, 这与 $\lim_{\Delta t \rightarrow 0} I_{i+1} = I_i$ 矛盾。所以, 定理中的条件①成立; 同理, 可证条件②成立。

定理 1 揭示了连续形变过程中构成网格图像的网格之间所存在的内在联系, 这种内在联系从微观的角度揭示出连续形变图像的网格图像之间的拓扑结构之间的关系, 并且得到了 $\lim_{\Delta t \rightarrow 0} I_{i+1} = I_i$ 的一种实现方法。定理 1 的重要意义还在于使得计算机从可能非常复杂的拓扑结构的分析和判断中解脱出来, 这种拓扑结构包括不同区域之间的邻接关系、嵌套关系等。但是, 在用定理 1 设计算法时, 需要进行两次循环, 分别对 G_i 和 G_{i+1} 进行判断, 对此加以进一步改进, 得到定理 2。

定理 2 I_{i+1} 是 I_i 的一次连续形变的充要条件是对任意小的 $\epsilon > 0$, 存在着有限个邻域 $W(I, s, \epsilon)$ ($s=0, 1, 2, \dots, S$), 使得

$$\textcircled{1} G_i \subseteq \bigcup_{s=0}^{s=S} W(I, s, \epsilon), G_{i+1} \subseteq \bigcup_{s=0}^{s=S} W(I, s, \epsilon);$$

$\textcircled{2} \forall_s (s=0, 1, 2, \dots, S)$, 存在 u, v , 使得网格 $(I, i, B) \in W(I+1, s, \epsilon)$, 网格 $(I+1, v, B) \in W(I, s, \epsilon)$ 。

其中, $u=0, 1, \dots, U-1, v=0, 1, \dots, V-1$ 。 U, V 分别是 G_i 和 G_{i+1} 中网格的个数。

事实上, 定理 2 是定理 1 的简单推论。定理 2 中的条件①、②成立时, 显然可以推出定理 1 中的条件①、②成立; 反过来, 当定理 1 中的条件①、②成立时, 取 $\epsilon = \max(r_1, r_2)$, 就可以得到定理 2 中的条件①、②。显然, 定理 2 的几何意义是明显的: 条件①表示了 G_i 和 G_{i+1} 可以同时被有限个邻域所覆盖, 条件②揭示了任意一个邻域内至少同时包含分别来自 G_i 和 G_{i+1} 中属性相同的两个网格。

3.2.2 追踪识别算法

以定理 2 中的邻域为圆域作为实例, 令圆的半径为 r , 下面把该圆域称为检测圆。图 3.20 给出了检测圆示意图。

本节追踪识别算法的总体结构为:

- (1) 求出当前帧的网格图像 G 。
- (2) 调用相邻两帧连续形变的跟踪算法, 若成功, 转下一步; 否则, 退出。
- (3) 是否为最后一帧? 若是, 退出; 否则, 转第(1)步。

网格图像的获取:

采用网格化技术可以快速得到图像的边界信息, 具体算法为:

- (1) 把已知图像等分成 $(M+1) \times (N+1)$ 个网格。
- (2) 依次求出每个网格的网格属性。
- (3) 只保留压线网格。

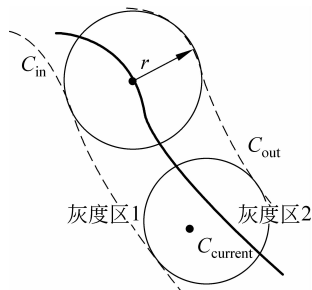


图 3.20 检测圆概念示意图

所得到的网格图像按从上到下、从左至右的顺序即按以行序为主的扫描顺序存放在数组中。

在应用实践中,对识别的精度要求往往有很大的差异,在许多情况下允许存在一定的误差转而要求有较高的响应速度,所以让用户来决定这种误差(即 M 和 N 的值)是可行的。用网格化技术代替逐点处理技术的另一个优点是算法的讨论以及数据结构的设计都带来很大的方便。应该注意到,从性质 1 可以看到,如果图像边界存在的话,则随着 M 和 N 取的越来越小,识别精度也越来越高。

3.2.3 相邻两帧连续形变的跟踪算法

假设上一帧和当前帧的网格图像已分别放在上一帧表和当前帧表两个表中,并已进行必要的初始化。

(1) 按顺序扫描上一帧表,设其当前指针为 p 。

(2) p 是否已超过上一帧表的表长? 若是,转第(6)步;否则,转第(3)步。

(3) 以 p 所指的网格坐标为中心建立检测圆,在当前帧的网格图像中统计落在检测圆内网格的个数 COUNTER。

(4) 如果 COUNTER=0,转第(7)步;否则,转第(5)步。

(5) $p \leftarrow p+1$, COUNTER=0, NO $\leftarrow$ NO+1,转第(2)步。

(6) 若 NO 等于当前帧表的网格个数时,则跟踪成功,退出;否则,转第(7)步。

(7) 跟踪失败,退出。

判断某网格 (I, x, B) 是否落在以矢量 P 为圆心、以 ϵ 为半径的检测圆内的方法是: 若

$$(x - P) \cdot (x - P) < \epsilon \quad (3.11)$$

则该网格落在检测圆内。

3.2.4 算法的进一步讨论

上述的算法是基于邻域为圆域而提出的。事实上,取定理中的邻域为八邻域,在算法的实现上也是很方便的。只需对 3.2.3 节的第(3)步稍加改进,即转而判断一个点是否在一个矩形内即可。也可以考虑其他不规整网格作为邻域,在一定条件下可以改变算法的精确度,但将影响算法的时间复杂度。

3.2.5 算法性能分析

不妨假设压线格的个数是 $M \times N$, 经过 $8 \times [M(N+1) + N(M+1)]$ 次比较就可得到所有网格的属性,从而获得压线格。又设检测圆的个数为 S , 则进行一次拓扑保持性连续形变算法识别需要进行比较的总次数是

$$T_G = 8[M(N+1) + N(M+1)] + S \quad (3.12)$$

可见,算法的时间复杂度只取决于对算法精度的要求,并且只与压线格的个数和检测圆的个数成一次线性关系。

3.2.6 实验结果

实验 1: 图 3.21 给出了连续形变的 Lena 图像序列。实验表明,取 $r=2$ 时可以得到理

想的跟踪结果。



图 3.21 连续形变的 Lena 图像序列及跟踪

实验 2: 图 3.22 是以 5ms 为时间间隔获取的手势录像帧序列。实验表明,本节算法可以准确地将它们识别为同一手势进行跟踪,且其最高精度可以达到 $r=1$ 。



图 3.22 以 5ms 为时间间隔获取的手势录像帧序列

实验 3: 图 3.23 是以 10ms 为时间间隔获取的手势录像帧序列。实验表明,本节算法可以准确地将它们识别为同一手势进行跟踪,且其最高精度可以达到 $r=2$ 。



图 3.23 以 10ms 为时间间隔获取的手势录像帧序列

3.3 对连续形变图像的追踪识别算法的再改进

3.3.1 基本性质和基本定理

首先给出以下性质:

性质 1: 网格图像细分的极限就是原图像的边界图,如果原图像的边界存在的话。

证明: 设 $G(k+1)$ 是 $G(k)$ 的一个细分,则

$$G(1) \supset G(2) \supset G(3) \supset \cdots \supset G(s) \supset \cdots \quad (3.13)$$

显然,如果原图像的边界图存在的话,则随着 s 的无限增大,上述序列必收敛,其极限就是原图的边界图。

性质 2: I_{i+1} 是 I_i 的一次连续形变的必要条件是: G_i 中所有网格的属性的集合 $= G_{i+1}$ 中所有网格的属性的集合。

证明: 设 I_{i+1} 是 I_i 的一次连续形变。据连续形变的定义, G_i 与 G_{i+1} 必同胚,从而 G_i 与 G_{i+1} 同构,顶点网格之间可以建立一一对应的关系,边图之间也有一一对应的关系,而这种一一对应都是基于网格属性的,所以结论成立。

性质 1 从理论上说明了用网格的手段来代替原图像边界的可行性以及在许可精度下的合理性。为了进一步揭示连续形变图像所对应的网格图像之间的内在联系,首先给出定理 1。

定理 1 I_{i+1} 是 I_i 的一次连续形变的充要条件是以下两个条件同时得到满足:

- ① 存在任意小的 $r_1 > 0$, 使得对 $\forall x \in G_i$, 存在 $y \in G_{i+1}$, 使得 $x \in W(I+1, y, r_1)$;
- ② 存在任意小的 $r_2 > 0$, 使得对 $\forall y \in G_{i+1}$, 存在 $x \in G_i$, 使得 $y \in W(I, x, r_2)$ 。

其中, x 和 y 的网格属性相同。

证明:

(1) 充分性的证明

首先注意到,若所述两个条件同时满足,则 G_i 与 G_{i+1} 必同胚。根据条件①, $\forall x \in G_i$, 存在 $y \in G_{i+1}$, 使得 $(I, p_x, B) \subseteq W(I+1, y, r_1)$, p_x 是 x 的网格坐标,所以 G_i 中属性的集合 $\subseteq G_{i+1}$ 中属性的集合;根据条件②, $\forall y \in G_{i+1}$, 存在 $x \in G_i$, 使得 $(I+1, p_y, B) \subseteq W(I, x, r_2)$, p_y 是 y 的网格坐标,所以 G_i 中属性的集合 $\supseteq G_{i+1}$ 中属性的集合,于是 G_i 中所有网格的属性的集合 $= G_{i+1}$ 中所有网格的属性的集合,从而网格图像 G_i 与 G_{i+1} 的顶点网格和边图之间可以建立一一对应的关系。另一方面, r_1 和 r_2 可以取任意小,当然可以远远小于不同的顶点网格和不同的边图之间的距离。在题设中的两个包含关系同时成立的条件下,必然保持有序相邻,否则所述的“两个包含关系同时成立”与“ r_1 和 r_2 任意小”之间将发生矛盾。

另外,若取 $r = \max(r_1, r_2)$, 则 G_i 和 G_{i+1} 都将夹在以 r 为厚度的条状区域内。即 $\lim_{\Delta t \rightarrow 0} I_{i+1} = I_i$ 成立。

(2) 必要性的证明

设 G_i 与 G_{i+1} 同胚且 $\lim_{\Delta t \rightarrow 0} I_{i+1} = I_i$ 。首先, G_i 和 G_{i+1} 的顶点网格之间可以建立一一对应的关系,所以,第一,若 x 是 G_i 的任意一个顶点网格,则必存在 G_{i+1} 中与 x 对应的顶点网格是 y , 并且 x 与 y 具有相同的网格属性,考虑到对任意小的 $\epsilon > 0$, G_i 和 G_{i+1} 都将夹在以 ϵ 为厚

度的条状区域内,所以 $|x-y|<\epsilon, x\subseteq W(I+1, y, \epsilon)$,这就证明了定理中的条件①;同理,可证条件②成立。第二,若 x 是 G_i 中的一个边图 $\langle h_{i,s1}, h_{i,s2} \rangle$ 上的任意一个网格,则在 G_{i+1} 中存在一个边图 $\langle h_{i+1,s1}, h_{i+1,s2} \rangle$ 与之对应。若存在 $r>0$ 和任意 $y\in G_{i+1}$,都有 $x\notin W(I+1, y, r)$,则 $|x-y|>\epsilon$,这与 $\lim_{\Delta t\rightarrow 0} I_{i+1} = I_i$ 矛盾。所以,定理中的条件①成立;同理,可证条件②成立。

定理1揭示了连续形变过程中构成网格图像的网格之间所存在的内在联系,这种内在联系从微观的角度揭示出连续形变图像的网格图像之间的拓扑结构之间的关系,并且得到了 $\lim_{\Delta t\rightarrow 0} I_{i+1} = I_i$ 的一种实现方法。定理1的重要意义还在于使得计算机从可能非常复杂的拓扑结构的分析和判断中解脱出来,这种拓扑结构包括不同区域之间的邻接关系、嵌套关系等。但是,在用定理1设计算法时,需要进行两次循环,分别对 G_i 和 G_{i+1} 进行判断,对此加以进一步改进,得到定理2。

定理2 I_{i+1} 是 I_i 的一次连续形变的充要条件是对任意小的 $\epsilon>0$,存在着有限个邻域 $W(I, s, \epsilon) (s=0, 1, 2, \dots, S)$,使得

$$\textcircled{1} G_i \subseteq \bigcup_{s=0}^{s=S} W(I, s, \epsilon), G_{i+1} \subseteq \bigcup_{s=0}^{s=S} W(I, s, \epsilon);$$

$$\textcircled{2} \forall s(s=0, 1, 2, \dots, S), \text{存在 } u, v, \text{使得网格 } (I, u, B) \in W(I+1, s, \epsilon), \text{网格 } (I+1, v, B) \in W(I, s, \epsilon).$$

其中, $u=0, 1, \dots, U-1, v=0, 1, \dots, V-1$ 。 U, V 分别是 G_i 和 G_{i+1} 中网格的个数。

事实上,定理2是定理1的简单推论。定理2中的条件①、②成立时,显然可以推出定理1中的条件①、②成立;反过来,当定理1中的条件①、②成立时,取 $\epsilon=\max(r_1, r_2)$,就可以得到定理2中的条件①、②。显然,定理2的几何意义是明显的:条件①表示了 G_i 和 G_{i+1} 可以同时被有限个邻域所覆盖,条件②揭示了任意一个邻域内至少同时包含分别来自 G_i 和 G_{i+1} 中的属性相同的两个网格。

3.3.2 追踪识别算法

以定理2中的邻域为圆域作为实例,令圆的半径为 r ,下面把该圆域称为检测圆。以下是总体识别算法:

- (1) 求出当前帧的网格图像 G 。
- (2) 调用相邻两帧连续形变的跟踪算法,若成功,转第(3)步;否则,退出。
- (3) 是否为最后一帧?若是,退出;否则,转第(1)步。

3.3.3 网格图像的获取

采用网格化技术可以快速得到图像的边界信息,具体算法为:

- (1) 把已知图像等分成 $(M+1)\times(N+1)$ 个网格。
- (2) 依次求出每个网格的网格属性。
- (3) 只保留压线网格。

所得到的网格图像按从上到下、从左至右的顺序即按以行序为主的扫描顺序存放在数组中。

在应用实践中,对识别的精度要求往往有很大的差异,在许多情况下允许存在一定的误

差转而要求有较高的响应速度,所以让用户来决定这种误差(即 M 和 N 的值)是可行的。用网格化技术代替逐点处理技术的另一个优点是算法的讨论以及数据结构的设计都带来很大的方便。应该注意到,从性质 1 可以看到,如果图像边界存在的话,则随着 M 和 N 取的越来越小,识别精度也越来越高。

3.3.4 相邻两帧连续形变的跟踪算法

假设上一帧和当前帧的网格图像已分别放在上一帧表和当前帧表两个表中,并已进行必要的初始化。

(1) 按顺序扫描上一帧表,设其当前指针分别为 p 。

(2) p 是否已超过上一帧表的表长? 若是,转第(6)步;否则,转第(3)步。

(3) 以 p 所指的网格坐标为中心建立检测圆,在当前帧的网格图像中统计落在检测圆内的网格个数 COUNTER。

(4) 如果 COUNTER=0,转第(7)步;否则,转第(5)步。

(5) $p \leftarrow p+1$, COUNTER=0, NO $\leftarrow$ NO+1,转第(2)步。

(6) 若 NO 等于当前帧表的网格个数时,则跟踪成功,退出;否则,转第(7)步。

(7) 跟踪失败,退出。

判断某网格 (I, x, B) 是否落在以矢量 P 为圆心、以 ϵ 为半径的检测圆内的方法是: 若

$$(x - P) \cdot (x - P) < \epsilon \quad (3.14)$$

则该网格落在检测圆内。

上述的算法是基于邻域为圆域而提出的。事实上,取定理中的邻域为八邻域,在算法的实现上也是很方便的。只需对 3.3.4 节中的第(3)步稍加改进,即转而判断一个点是否在一个矩形内即可。也可以考虑其他不规整网格作为邻域,在一定条件下可以改变算法的精确度,但将影响算法的时间复杂度。

3.3.5 算法性能分析

不妨假设压线格的个数是 $M \times N$, 经过 $8 \times [M(N+1) + N(M+1)]$ 次比较就可得到所有网格的属性,从而获得压线格。又设检测圆的个数为 S , 则进行一次拓扑保持性连续形变算法识别需要进行比较的总次数是

$$T_G = 8[M(N+1) + N(M+1)] + S \quad (3.15)$$

可见,算法的时间复杂度只取决于对算法精度的要求,并且只与压线格的个数和检测圆的个数成一次线性关系。

另外,进一步将本书提出的算法与文献[159]所给算法进行比较。在文献[159]中,每次识别的时间复杂度为 $O(s \times n)$, 其中 s 为物体中圈的数量, n 为每个圈的顶点个数,其中 $s \times n$ 表示与源图中顶点的总数,在一定的条件下,时间复杂度可以得到有效改善。另外,原算法与源图的拓扑结构有关,而这里与源图的拓扑结构无关,这正是本算法的又一个重要特点。

虚拟现实技术是一种高度逼真的模拟人在自然环境中视、听、动等行为的人机交互技术,是交互——沉浸——想象力的总集成。在人机交互技术中,典型的做法是通过摄像机对人体扫描,得到扫描图片,从中获取人体姿态,再根据不同的应用需要提取所感兴趣的信息,这些信息包括人体运动姿态信息、手势信息、面部表情信息等。事实上基于以下两个原因,

这些不同的算法可以得到有机的统一：

- (1) 相邻两帧图片可以认为存在连续形变关系。
- (2) 都需要对图像照片进行锁定性跟踪识别。

例如,在基于手势的交互系统中,往往根据伸直手指的数目对手势进行分类和识别,但目前大多数系统都采用刚性识别算法,其严格的限制影响操作者的沉浸感和交互效果。

3.3.6 实验结果

实例 1: 图 3.21 给出了连续形变的 Lena 图像序列,实验表明,取 $r=2$ 时可以得到理想的跟踪结果。

实例 2: 图 3.24 是以 100ms 为时间间隔获取的一流体图像序列,实验表明,其最高精度可以达到 $r=1$ 。

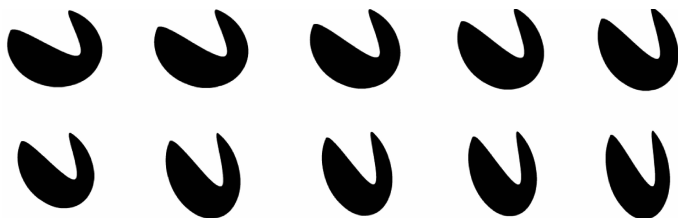


图 3.24 以 100ms 为间隔获取的流体图像序列及跟踪

连续形变物体的追踪算法研究要解决的关键问题是：

- (1) 怎样判断被追踪物体的拓扑结构没有改变,或者这种改变在允许的误差范围内;
- (2) 对连续形变“渐变性”的识别。

从拓扑结构理论的角度考察,拓扑结构的判断是一个相当复杂的问题。本研究把这两个问题结合起来分析和研究,通过引入网格、网格属性等基本概念,从理论上揭示出了连续形变与拓扑结构和渐变性之间所存在的深刻联系,建立了一套基于网格等基本概念的、以定理 1 为代表的基本理论。

以之为基础设计的算法具有以下显著特点：

- (1) 避开了复杂的拓扑结构分析;
- (2) 识别的精度可以根据不同的应用需要加以灵活调节;
- (3) 算法简单,实现方便;
- (4) 算法的时间复杂度与图像的拓扑结构的复杂程度无关;
- (5) 算法的灵活性、普适性和快速性,使之特别适合人机交互系统的设计;
- (6) 算法对被追踪物体的拓扑结构、灰度以及色彩等可以不加任何限制。

算法的一个主要弊端是对被跟踪物体的图像噪声(尤其是高光)很敏感。

3.4 基于机器学习和人工神经网络技术的手势识别

通过机器学习的方法来获取高维特征集与手势之间的关系是解决高维特征空间搜索问题的一个有效方法,HMM 就是其中的一种重要方法,它可以有效地提取和识别手势的时空动态模型^[165-167]。一个 HMM 由大量的隐含状态所组成,每个状态有一个由它自己到其他

状态的转移概率,转移概率被建模为 n 阶 Markov 过程(即转移概率仅仅取决于在该状态之前的 n 个状态)。HMM 的实质在于量化时间(观测)序列,使之变为少量的离散状态,且在状态之间(包括自己)具有转换概率。这种结构的一个重要特点是 HMM 是一种高度的时间尺度(Time-Scale)不变量,在每个时间步上都将从当前 HMM 状态输出一个观测特征(例如人手运动方向等),由参数密度函数表示其概率分布。在训练阶段,获取 HMM 各种参数,例如转移概率,输出概率密度函数。HMM 可以找到最可能产生已知量测序列的状态序列^[58]。

在文献[168]中,通过特征提取、训练和识别构成一个基于模型的手势识别系统。在特征提取阶段,结合空间(边界)和时间(运动)信息提取图像特征;在训练阶段,使用主成分分析法(Principal Component Analysis, PCA)刻画空间形状变化,采用 HMM 刻画时间形状变化,使用改进的 Hausdorff 距离测度来度量特征图像和事先存储的 PCA 模型之间的相似性;在识别阶段,利用事先学习的 PCA 模型和 HMM,从输入图像序列产生测量模型,然后用 Viterbi 算法^[169]确定手势。实验表明,该方法可以识别 18 种连续变化的手势。

文献[51]提出一种随机的、非线性监督学习模型 SMA (Specialized Mappings Architecture)。SMA 方法采用“分而治之”策略^[170]减少问题空间的复杂度。SMA 由几个特定的从训练集数据通过学习可以自动得到的前向映射函数和一个已知的反馈函数所组成,每个特定函数把输入空间(例如图像特征空间)的某些域映射到输出空间之上(例如人手关节参数),可以比较好地处理病态问题。

文献[134]用一组手势训练集作为 Fuzzy C-Means Clustering 算法的输入,以 12 个基本手势作为模型(Pattern)。文献[171]中,对可形变模型进行无监督学习。根据模型形变量的统计信息,可以通过优化技术估计形变量。文献[19]中提出一种基于神经网络的连续手势识别系统,首先结合先验知识的人工神经网络层次化系统定位图像中指尖的 2D 位置,然后有一个神经网络系统把该 2D 位置转化为 3D 手模型,该模型根据自然手的维数以及运动概率进行设计。

3.4.1 HMM 原理

由于 HMM 在手势识别方面取得了很大成功,下面简要介绍 HMM 的原理。

设随机过程为 $X(t)$,在每一时刻 $t_i (i=1,2,\dots)$, $X_n = X(t_n)$ 可以处在状态 $q_0, q_1, \dots, q_{N-1}$ 之一,而且只在 $t_1, t_2, \dots$ 时刻发生状态转移。在这种情况下,若 t_{i+k} 时刻变成任一状态 $q_u (u=0,1,\dots,N-1)$ 的概率,只与该过程在 t_i 时刻的状态有关,而与 t_i 时刻以前过程所处的状态无关,则称这个过程为 Markov 链,即满足

$$P(X_{i+k} = q_r \mid X_0 \cdots X_i = q_w) = P(X_{i+k} = q_r \mid X_i = q_w) \quad (3.16)$$

其中, $i, k=1,2,\dots$ 。

HMM 是在 Markov 链基础上发展而来的,一个 HMM 可以定义为^[172]

$$\lambda = (A, B, \pi) \quad (3.17)$$

它主要有 5 个概率参数: A, B, N, M, π 。

(1) 内部状态集合: $Q = \{q_0, q_1, \dots, q_{N-1}\}$, $|Q| = N$, N 为 HMM 模型中 Markov 链的状态数目。

观察集合: $V = \{v_0, v_1, \dots, v_{M-1}\}$, $|V| = M$, M 表示每个状态可能对应的观察数目。

(2) 状态转移概率矩阵:

$$A = \{a_{ij}\}$$

$$a_{ij} = P(X_{t+1} = q_i | X_t = q_j)$$

(3) 观察概率矩阵:

$$B = \{b_{jk}\}$$

$$b_{jk} = P(O_t = v_k | X_t = q_j)$$

(4) 观察序列:

$$O = \{O_0, O_1, \dots, O_{T-1}\}, \quad O_i \in V, i = 0, 1, \dots, T-1, |O| = T$$

(5) 初始概率矢量: $\pi = \{\pi_0, \pi_1, \dots, \pi_{N-1}\}$

这里

$$\sum_{i=0}^{N-1} \pi_i = 1 \quad (3.18)$$

HMM 问题就是在给定 HMM 参数 $\lambda = (A, B, \pi)$ 的条件下, 求观察序列 $O = \{O_0, O_1, \dots, O_{T-1}\}$ 的产生概率 $P(O|\lambda)$ 。

设隐状态序列为 $X = (X_0, X_1, \dots, X_{T-1})$, 由全概率公式可以得到

$$P(O|\lambda) = \sum_X P(O, X|\lambda) = \sum_X P(O|X, \lambda) P(X|\lambda)$$

其中

$$P(O|X, \lambda) = b_{i_0}(O_0)b_{i_1}(O_1)\cdots b_{i_{T-1}}(O_{T-1})$$

所以

$$P(O|\lambda) = \sum_X [\pi_{i_0} b_{i_0}(O_0)][a_{i_0 i_1} b_{i_1}(O_1)] \cdots [a_{i_{T-2} i_{T-1}} b_{i_{T-1}}(O_{T-1})]$$

图 3.25 进一步给出了一个 HMM 示意图。同 Markov 链相比, HMM 描述了更为复杂的随机过程, 之所以称为隐马尔可夫模型, 是因为它所描述的系统无法从外部直接观察到内部的状态, 即观察集合不同于状态集合。一个 Markov 链描述了内部状态的转移规律, 另一个随机过程则描述了内部状态与外部观察值之间的统计对应关系。也就是说, 隐马尔可夫过程是一个双重的随机过程: 一个潜在的过程称为“状态”过程, 另一个可观测过程称为“观测序列”, 观测序列是由隐含的状态过程决定的。所以, 一个 HMM 也可以用图 3.26 进行刻画。

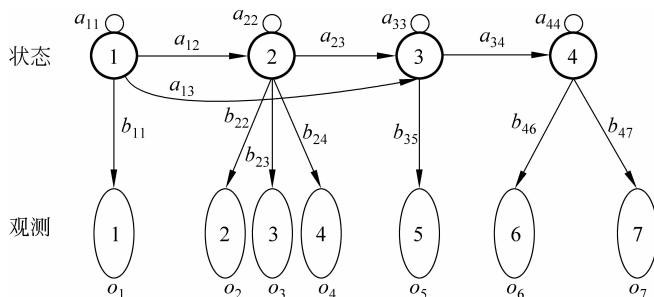


图 3.25 HMM 概念示意图