基于 Hausdorff 距离的离散点集模糊模式匹配

邢伟　006104　吴学文 006103　伍瑞　006125

摘要

本文讨论了模式匹配的度量问题，提出了基于 Hausdorff 距离的离散点集模糊模式匹配的模型，并对此模型进行了论述，然后列举了需要解决的问题。在此基础上讨论解决此问题的已有算法。最后，对程序实际采用的算法进行了详细的叙述。

关键词

模式匹配(Pattern Matching)　Hausdorff 距离　近似最近邻点查找

正文

一．　引言

模式匹配作为人工智能的一个研究领域，已经获得了很大的发展，而模式匹配的度量尺度则是影响匹配效果的一个重要因素，因而这方面存在许多有意思的研究课题。

我们找到的"approximate Geometric Pattern Matching under Rigid Motions"这篇文章，对模式匹配的度量尺度进行了探讨，提出了基于 Hausdorff 距离的模式匹配原型算法，并验证了其可行性。我们在对这篇文章和其它参考文献的详尽理解的基础上，对这篇文章所提到的算法进行了一定的改进，并且完成了程序实现。

我们考虑的模式匹配是基于离散点集的。我们将先给出该算法的理论依据，然后介绍该算法的实现，最后研究试验结果，并得出结论。

二．　背景

1．离散点集模式匹配

假设我们给定 D 维空间的两个点集：具有 n 个点的 B（背景）及具有 m 个点的 P（模式）。基于几何的模式匹配方法就是要寻找到一条刚性的运动轨迹，使当 P 沿这条刚性轨迹运动后，P 的每个点对应 B 中的相应点。

最简单的解决方法就是逐点进行精确匹配，但是这种方法对噪声异常敏感。实际应用中更为使用的是近似匹配，即寻找到 P 的一条刚性的运动路径，使 P 的每个点都接近 B 中的某个点，通常我们令刚性运动为 T。

2．以 Hausdorff 距离为度量尺度

两个离散点集的 Hausdorff 距离定义如下：

设两个点集分别为 C，D，则从 C 到 D 的 Hausdorff 距离 h(C,D) = $\max_{c \in C} \min_{d \in D} p(c,d)$，p 通常是欧式距离，注意 h(C,D)通常不等于 h(D,C)。这就说明对于 C 中每个点来说，与 D 中对应点的误差都在 h(C,D)范围内。使用 Hausdorff 距离作为匹配度量，就可以在背景上查找模式。

三． 算法思想

本文算法的基本思想就是要找到刚性运动轨迹，使模式点集与背景点集中的某一部分近似匹配。即给定 D 维点集 P，B，要求求出一个刚性运动 T，使得 h(T(P),B)最小。由于这种方法是基于 Hausdorff 距离进行度量的，因此对于背景有噪声的情况，可以获得很好的结果。

本算法并非精确匹配，而是近似匹配，即保证所求得的 T 可以使 $h(T(P),B) \le \alpha\varepsilon^{*}$，其中 ε^{*} 是最优值，α 是大于 1 的某个小的常数。

下面介绍基于 Hausdorff 距离模式匹配的原型算法。

1．D 维空间的纯平移

设平移为 T，我们所要求的是这样一个 $T^{'}$，即 $h(T^{'}(P),B)$ 最多等于 $2h(T_{opt}(P),B)$，其中 T_{opt} 是最优旋转，即 $h(T_{opt}(P),B)$ 最小。

基本思路是这样的：从 P 中选择某个点 p 作为代表点。对于每个$b \in B$，定义T_b为使 p 移到 b 的平移。我们的目标就是要寻找到$\min_{b \in B}\{h(T_b(P), B)\}$。

设这样的T_b为$T^{'}$，则在二维空间有$h(T^{'}(P), B) \le 2h(T_{opt}(P), B)$，在 D 维空间有$h(T'(P), B) \le (2+\varepsilon)h(T_{opt}(P), B)$，其中$\varepsilon \in (0,1)$。

证明：

（1） 二维情况

设$h_{opt} = h(T_{opt}(P), B)$，对于每个$p \in T_{opt}(P)$，存在某个相关的$b \in B$，与 p 的距离小于等于$h_{opt}$。当对$T_{opt}(P)$进行平移以使 p 与 b 重合时，则最多只要移动$h_{opt}$即可，此时任意 p 在 B 中的最邻近点与 p 的距离最多不超过$2h_{opt}$。

（2） D 维情况

对于 D 维空间，为了降低算法复杂度，本算法采取 Arya et al.的近似最近邻查找算法，该算法导致 T(P)有可能比实际值大$1+\varepsilon^{'}$倍。再由二维情况的结论，可知任意任意 p 在 B 中的最邻近点与 p 的距离最多不超过$2(1+\varepsilon^{'})h_{opt}$。

2．二维空间中的平移及旋转

对于同一平面上的点，我们可以找到一个$E^{'}$，使$h(E^{'}(P), B)$小于等于$4h(E_{opt}(P), B)$，其中 E 为欧式运动，E_{opt}为最优欧式运动。

基本思路是这样的：从模式中取出相距最远的两个点 r，k；这可以在$O(m^{2})$时间内完成。、r 既作为平移中的代表点，又作为旋转的中心。对于每个$b \in B$，令T_b为将 r 移到 b 的平移变换。对于每个$b^{'} \in B, b^{'} \ne b$，令$R_{b^{'}}$为使 r，$b^{'}$,k 共线的旋转变换。令$E_{b,b^{'}}$ 为$T_b R_{b^{'}}$。则我们的目标是找出$\min_{b,b^{'} \in B}\{h(E_{b,b^{'}}(P), B)\}$。

这样做有$h(E^{'}(P), B) \le (4+\varepsilon)h(E_{opt}(P), B)$，对于任何常数$0 \le \varepsilon < 1$成立。

证明：令$h_{opt} = h(E_{opt}(P), B)$，对于任意$p \in E_{opt}(P)$，存在对应的$b \in B$，与 p 的距离小于等于$h_{opt}$。由上面所说，为使 p 与 b 重合，最多会令 p 与其在 B 重的最

邻近点的距离增加h_{opt}。考虑旋转情况，由于这使 k 最多移动$2h_{opt}$，且 k 是模式中与 r 距离最远的点，因此模式中其他的点移动最多不超过$2h_{opt}$，所以结果最多为$4h_{opt}$，又由于采用上面所介绍的最近邻查找法，因此最多不超过$4(1+\varepsilon)h_{opt}$。

3．三维空间中的旋转

对于三维空间中的点，可以找到一个关于原点的旋转变换$R^{'}$，使$h(R^{'}(P),B)$小于等于$4h(R_{opt}(P),B)$。

基本思路如下：首先在模式中找到距离原点最远的点 p1。然后找到与 p1 及原点连线垂直距离最远的点 p2，对于任意$b^{'}\in B$，定义$R1_{b^{'}}$为使原点，p1 及 b’共线的旋转变换，对于任意$b^{''}\in B$，$b^{''}\neq b^{'}$，定义$R2_{b^{''}}$为使以原点和 p1 连县为轴，原点，p1，p2 和$b^{''}$共面的旋转变换。我们的目的是找到$\min_{b^{'},b^{''}\in B}\{h(R2_{b^{''}}(R1_{b^{'}}(P)),B)\}$。

由上述方法有$h(R'(P),B)\le(4+\varepsilon)h(R_{opt}(P),B)$，对于任意常数$0\le\varepsilon<1$成立。

证明：

令$h_{opt}=h(R_{opt}(P),B)$，对于任意$p\in R_{opt}(P)$，存在$b\in B$与 p 相距h_{opt}以内。通过旋转使 p1 与 b 及原点共线，这使模式中每个点最多移动h_{opt}。而$R2_{b^{''}}$最多使点移动$2h_{opt}$，所以最多增加$3h_{opt}$。所以模式中每个点与最近林背景点最多相距$4h_{opt}$。

4．三维空间中的平移及旋转

对于三维空间中的点，我们要找出一个E'，使$h(E'(P),B)$最多为$(8+\varepsilon)h(E_{opt}(P),B)$。

方法如下：从模式中选择相距最远的两点 r，k。选择 l，使之与直线 rk 的垂直距离最大。对于任意$b\in B$，定义T_b为将 r 移到 b 的平移变换。对于任意

$b' \in B, b' \neq b$，定义 $R1_{b'}$ 为使 r，k 及 b’共线的旋转变换，对于任意 $b'' \in B, b'' \neq b', b'' \neq b$，定义为以 rk 连线为轴，r，k，l 和 b'' 共面的旋转变换。则目的就是求出 $\min_{b,b',b'' \in B} \{h(R2_{b''}(R1_{b'}(T_b(P))), B)\}$。

由上述方法，有 $h(E'(P), B) \le (8+\varepsilon) h(E_{opt}(P), B)$，对于任意常数 $0 \le \varepsilon < 1$ 成立。

证明：

令 $h_{opt} = h(E_{opt}(P), B)$，对于任意 $p \in E_{opt}(P)$，存在 $b \in B$ 与 p 相距 h_{opt} 以内。对于平移变换，模式中每个点最多移动 h_{opt}。第一次旋转变换使模式中每个点最多移动 $2h_{opt}$，第二次旋转变换使模式中每个点最多移动 $4h_{opt}$。所以最多距离 $8h_{opt}$。

四. 实现算法

算法步骤

我们详细的阅读了描述该问题的论文，并查阅了与此相关的一些资料，完成了论文中所描述的算法，并且通过验证，达到了比较令人满意的模式匹配效果。

我们的算法步骤大致分为三部分：

第一部分 对输入的背景和模板图像进行分析，提取出轮廓并将之转化为离散的特征点集；

第二部分 把模板的特征点集在对应的二维或三维空间中进行平移和旋转，求得特征点集变换后的坐标；

第三部分 对每一次变换后的模板点集，与背景点集进行 hausdorff 距离的计算，这一计算是通过对近似最近邻点的查找来完成的；

最后，将所记录的最小 hausdorff 距离时的位置作为最佳模糊匹配，即完成查找。

实现难点

1. 对图像之类的位图数据进行模式匹配时，需要对其进行特征点提取。
2. 坐标转换过程中，需要实现三维点集绕任意轴旋转的坐标变换。
3. 求 hausdorff 距离，需要解决点在点集中的最近邻点的查找。

详细算法

对于二维空间的模糊模式匹配，我们需要考虑背景及模板图像特征点集的提取，我们主要采用的是将 RGB 图像转化为灰度图像，然后进行二值处理，在强化边缘的同时生成二维特征点集。

对于三维空间的特征点集生成，因为实现算法的时间缘故，我们暂时只是直接生成了背景与模板的三维特征点集，以供测试使用。我们实现了三维特征点集的存取功能，即能对生成的特征点集存入磁盘，以及从磁盘中将事前生成好的特征点集取出。

对于模板特征点集在刚性条件下的移动，主要有平移和旋转两种方式。文献[1]在模糊模式匹配的过程中主要考虑了四种方式的坐标转换：二维空间的纯平移、二维空间的平移和旋转、三维空间的纯旋转、三维空间的平移和旋转。从实用角度出发，我们把这四种情况综合成了两种转换方式，其中：

二维空间的转换，包括平移和旋转。首先，在模板的特征点集中寻找相距最远的两点 r 和 k，将其中一点作为平移时的参照点和旋转时的中心点。对于背景点集中的每一点 b，先将模板点集以 r 为参照点平移至 b，然后对于背景点集中除 b 外的每一点 b’，将模板点集以 r 为中心点进行旋转，使得 r、b’、k 共线，求得这时模板点集与背景点集的 hausdorff 距离。最小值者即为最佳的模糊模式匹配，在求 hausdorff 距离过程中，应用文献[2]介绍的近似最近邻点查找算法，可使得整体转换算法的时间复杂度降至 $O(n^2 m \log n)$ 。

三维空间的转换要复杂一些，但同样是只包括平移和旋转两种变换方式。类似的，首先在模板的三维征点集中寻找相距最远的两点 r 和 k，再选出距直线 rk 最远的一点 l，并将 r、k 其中一点作为平移时的参照点和旋转时的中心点。对于背景点集中的每一点 b，先将模板点集以 r 为参照点平移至 b，第二重循环与二

维时一样，对于背景点集中除 b 外的每一点 b’，将模板点集以 r 为中心点进行旋转，使得 r、b’、k 共线，之后进行第三重循环，对于背景点集中除 b、b’外的每一点 b’’，将模板点集以直线 rk 为轴进行旋转，使得 b’’、r、k、l 共面，求得这时的模板点集与背景点集的 hausdorff 距离。最小值者即为最佳的模糊模式匹配。在求 hausdorff 距离过程中，应用文献[2]介绍的近似最近邻点查找算法，可使得整体转换算法的时间复杂度降至 $O(n^3 m \log n)$。

在求 hausdorff 距离过程中，对于模板点集中的每一点，求出它在背景点集中的最近邻点，其中的最大距离即为 hausdorff 距离。在参考了文献[3]介绍的几种最近邻点查找算法以后，我们采用了文献[2]介绍的一种近似最近邻点查找算法，主要原因是其有效的降低了算法的时间复杂度，并且二维和三维的最近邻点查找均可用同一种算法来实现。

该算法主要应用了一种被称为 BD_tree 的数据结构来对被查找点集的空间进行分割，建立 BD 树后，其叶子节点为一系列的子空间，每个子空间含有点集中不超过一个的点，然后再通过对查找点进行 BD_tree 的搜索来找出其近似的最近邻点，其误差不超过给定的甚小正数 e。具体过程如下：

首先是对给定的被查找点集，建立相应的 BD 树，对空间进行划分。、主要目的是为了实现沿树的路径下降时，其子空间的大小和所包含点数均呈指数下降，从而保证树的最大深度不超过 log *n*，我们对点集的分割分别应用了 split 和 shrink 两种分割方法，其中 split 分割是在子空间跨度最大的维度上，进行在中点的剖分，可以保证沿树的路径下降时，其子空间的大小呈指数下降；而 shrink 方法则在子空间内部，点聚集的比较密集的地方进行小区域的剖分，使得分割后的内外两空间所含点数均不超过某一比例，因此保证沿 BD 树路径下降时，其子空间所包含点数均呈指数下降。考虑到算法的具体实现，我们总是优先采用 split 对空间进行分割，如果不能满足沿深度指数下降的条件，再进行 shrink 分割。重复剖分过程，直到每一子空间只含有不超过一个点而形成 BD 树的叶子节点为止。

第二步，对于查找点 q，首先确定其位于哪一个子空间里，将其余的各个子空间与查找点的距离按从小到大排序，这是通过对 BD 树的优先搜索并建立优先搜索队列而得到的，首先把 BD 树根节点入队列，然后按其子空间到查找点的最

近距离进行 BD 树的深度搜索，同时依次把其兄弟节点插入到队列的相应位置，使队列始终保持按到 q 的距离从小到大排列。对于依次出队列得到的子空间，计算其所包含点与查找点的距离，我们记录最近距离的点为 p，则当前子空间与查找点的距离 d>dist(p,q)*(1+e)时终止循环，则点 p 即为所求得近似最近邻点，其误差不超过 e。

五．具体实现

在上面讨论的基础上，我们进行了程序实现，采用了 Visual C++ 6.0 作为实现工具。所用的机器的配置为 Celeron-450，内存 256M。

软件功能说明

在我们实现的软件中，主要包括如下的几个功能：

1．二维图像特征点提取功能：采取轮廓提取算法提取特征点；

2．二维背景点集，模式点集生成功能：可以将二维图像读入，并将特征点存入点集中；

3．模式匹配功能：进行模式匹配。

六．数据结果分析及讨论

通过程序的实验结果，我们可以发现，当背景点集存在噪声时，对匹配精度影响较小，匹配结果基本令人满意，但是当模式点集存在噪声时，会对匹配结果产生很大影响，因此应该在匹配之前，对模式点集进行滤波，以尽量消除噪声。

对模式点集进行小的扰动后，可以看到匹配结果仍然可以接受，这说明本算法的抗干扰性还是很好的。

但是，本算法的速度还不能令人满意，对于背景点集和模式点集的规模都很小的时候，速度还可以令人接受，但是随着两个点集规模的增大，其计算时间迅速增加。这说明该算法还有很大的冗余，需要进行改进。

需要改进的地方我们认为大概应有以下几点：

1．在求 Hausdorff 距离时，可以用一定的域值来限定可接受值的范围，当所求得的 Hausdorff 距离小于该域值时，认为该结果可以接受，这样就可以

结束计算。这样做不会减少最坏情况的计算复杂度，但可以显著减少期望复杂度。问题是域值如何取。一个可行的方法是进行学习。

2. 在对模式中的每个点求其在背景中的最近邻点，以确定模式与背景的 Hausdorff 距离时，会进行一系列比较。当已知前一点与背景点集的 Hausdorff 距离时，则在求当前点在背景中的最近邻点所进行的比较中如果发现背景中有一点与当前点的距离小于前一点与背景点集的 Hausdorff 距离，就可以终止该点的最近邻查找，因为该点与背景点集的 Hausdorff 距离已不是最大距离了。

参考文献

[1] Michanel T.Goodrich, Joseph S.B.Mitchell, Mark W.Orletsky. Approximate Geometric Matching Under Rigid motion.

[2] Sunil Arya, David M.Mount, Nathan S.Netanyahu, Ruth Silverman, Angela Y.Wu. An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions.

[3] 周培德 《计算几何》 清华大学出版社 2000