

一种利用 Hausdorff 距离的高效目标搜索算法

彭晓明 丁明跃 周成平 张天序

(华中科技大学图像识别与人工智能研究所, 图像信息处理与智能控制教育部重点实验室, 武汉 430074)

摘要 为了快速高效地进行目标搜索, 提出了一种在仿射变换条件下, 利用 Hausdorff 距离进行目标搜索的高效算法。此算法是在一种新的距离变换形式——“最小正方形盒距离变换”的基础上进行的。实验结果表明, 与现有算法相比, 该算法在不影响搜索成功率和目标定位精度的情况下, 还可以显著地缩短搜索时间。为验证该算法的有效性, 将该算法与 Rucklidge 算法进行了对比实验, 结果表明, 该算法明显优于 Rucklidge 提出的快速目标搜索算法。

关键词 目标定位 模式识别 模板匹配 Hausdorff 距离

中图分类号: TP391.4 文献标识码: A 文章编号: 1006-8961(2004)01-0023-06

A Highly Efficient Approach to Locating Objects Using the Hausdorff Distance

PENG Xiao-ming, DING Ming-yue, ZHOU Cheng-ping, ZHANG Tian-xu

(Institute for Pattern Recognition and Artificial Intelligence, State Education Commission Key Lab for Image Processing and Intelligent Control, Huazhong University of Science and Technology, Wuhan 430074)

Abstract The Hausdorff distance, a measure defined between two point sets, has been used to search for objects in images for its robustness and reliability. In this paper, a highly efficient approach, which uses the Hausdorff distance to locate an object that has undergone an affine transformation, is presented. Prior to our study, the most efficient methods for solving such a problem are two of Rucklidge's methods: "without zoning" and "4×4 zoning". The proposed approach shares some similarities with Rucklidge's methods but is mainly based on the "smallest square box distance transform" introduced in this paper. The idea of "smallest square box distance transform" is described in detail in this paper along with its implementation. By replacing the "box distance transform" of Rucklidge's methods with our "smallest square box distance transform" in the search process, a much higher search speed can be achieved while maintaining the search success rate and target location accuracy. In all, two experiments are given in this paper. One experiment is to test the proposed approach on edge images and the other experiment is based on feature point images. It can be seen from the experimental results of both experiments that the proposed approach apparently outperforms Rucklidge's fast methods.

Keywords Object location, Pattern recognition, Model matching, the Hausdorff distance

1 引言

Hausdorff 距离作为一种距离测度, 常用于衡量两个二维点集之间的相似程度。由于使用 Hausdorff 距离作为距离测度时无需考虑两个点集中的点与点之间的对应关系, 因此可以有效解决当图像中存在噪声和目标被部分遮挡时的识别问题。

利用 Hausdorff 距离进行目标搜索, 其关键是采用高效的搜索策略。在文献[1]中, Huttenlocher 等人首先提出了一种用于搜索图像中经过平移变换的目标模板的快速算法, 由于平移变换只包括两个参数, 所以该算法的搜索空间是二维的。实践证明, 这种算法经过补充后, 可以搜索经过平移和旋转后的目标模板; 随后, 在 Huttenlocher 和 Rucklidge 的技术报告^[2]中又提出了一种搜索空间是四维的目标

搜索算法,更可以快速搜索在水平和垂直两个方向上分别同时发生变比和平移后的目标模板,与文献[1]中的二维空间快速搜索算法相比,此算法要相对复杂得多;此后 Rucklidge 又提出了一种用于搜索经过仿射变换后的目标模板的算法^[3]。由于仿射变换包括 6 个参数,所以算法的搜索空间相应是 6 维的。由于仿射变换包括前面所提到的所有形式的变换,故而 Rucklidge 的算法适用于前面所提到的所有目标搜索任务。在 Rucklidge 之后,目前尚没有人提出更有效的搜索算法来解决此类仿射变换下的目标搜索问题。本文在 Rucklidge 算法的基础上,提出了一种更为有效的搜索算法,实践证明,新算法可以大大缩短所需的搜索时间。

2 Hausdorff 距离

以 I 和 M 分别代表搜索图像(以下简称图像)和目标模板(以下简称模板),它们是经过特征抽取后(例如经过边缘提取)生成的二维点集, I 和 M 之间的 Hausdorff 距离 $D_{\text{Hausdorff}}(M, I)$ 可定义为

$$D_{\text{Hausdorff}}(M, I) = \max(d_{\text{Hausdorff}_F}(M, I), d_{\text{Hausdorff}_R}(I, M)) \quad (1)$$

其中, $d_{\text{Hausdorff}_F}(M, I)$ 和 $d_{\text{Hausdorff}_R}(I, M)$ 分别为前向 Hausdorff 距离和后向 Hausdorff 距离,其定义分别为

$$\begin{aligned} d_{\text{Hausdorff}_F}(I, M) &= \max_{m \in M} \min_{i \in I} \|m - i\| \\ d_{\text{Hausdorff}_R}(M, I) &= \max_{i \in I} \min_{m \in M} \|i - m\| \end{aligned} \quad (2)$$

式(2)中的 $\|\cdot\|$ 为距离范数,这里采用 L_2 范数。由于在实际应用中存在噪声,同时可能出现目标被部分遮挡的情况,因此常用部分 Hausdorff 距离来代替式(1)和式(2)中的 Hausdorff 距离。

$$D_{\text{Hausdorff}}^{(f_F, f_R)}(M, I) = \max(d_{\text{Hausdorff}_F}^{(f_F)}(M, I), d_{\text{Hausdorff}_R}^{(f_R)}(I, M))$$

$$D_{\text{Hausdorff}_F}^{(f_F)}(M, I) = f_F \text{th} \min_{m \in M} \|m - i\| \quad (3)$$

$$D_{\text{Hausdorff}_R}^{(f_R)}(I, M) = f_R \text{th} \min_{i \in I} \|i - m\|$$

式(3)中, f_F 和 f_R 分别为前向和后向 Hausdorff 分数,且有 $0 < f_F, f_R \leq 1$ 。 $f \text{th} g(x)$ 形式的含义为:给定一个取值范围在 $(0, 1]$ 之间的分数 f , 求在集合 X 上位于分位数 $f \text{th}$ 处的 $g(x)$ 值。例如,当 $f=1$ 时(对应 1th), 结果为 $g(x)$ 的最大值; $f=1/2$ 时(对应 1/2 th), 结果为 $g(x)$ 的中值。设 n_M 和 n_I 分别表示 M 和

I 中的非零点(特征点)数目。下面以 $d_{\text{Hausdorff}_F}^{(f_F)}$ 为例对式(3)进行说明:对任意一个模板点 $m \in M$, 可以求得一个 $\min_{i \in I} \|m - i\|$ 值。对所有模板中的这些 $\min_{i \in I} \|m - i\|$ 值(共有 n_M 个)按照升序(从小到大)排列,那么处在该排列中第 $f_F n_M$ 个位置的数值即为 $d_{\text{Hausdorff}_F}^{(f_F)}$ 。同理可理解 $d_{\text{Hausdorff}_R}^{(f_R)}$ 。此外,由于通常 M 的尺寸要小于 I , 也就是说,目标可以完全包容于搜索图像中,所以,在计算 $d_{\text{Hausdorff}_R}^{(f_R)}(I, M)$ 时,并不考虑 I 中的所有像素点,而是只考虑 I 中目标所占区域 I_{box} 中的那一部分像素点,即

$$d_{\text{Hausdorff}_R}^{(f_R)}(I, M) = f_R \text{th} \min_{(i_x, i_y) \in I_{\text{box}}} \|m - (i_x, i_y)\| \quad (4)$$

使用式(4)时,假定目标被包容于图像 I 中 $[x_{\min}, x_{\max}] \cup [y_{\min}, y_{\max}]$ 所限定的一个方盒(box) I_{box} 中,其中, $x_{\min}, x_{\max}, y_{\min}$ 和 y_{\max} 分别为 I_{box} 的坐标取值范围的上、下限; $n_{I_{\text{box}}}$ 表示 I_{box} 中的非零点数目,则仿射变换的公式可定义为

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (5)$$

式(5)中的 $a_{00} \sim t_y$ 为仿射变换的参数。为简单起见,可以用这些参数组成一个 6 元组 $t = (a_{00}, a_{01}, a_{10}, a_{11}, t_x, t_y)$ 来代表一个仿射变换。在 t 的作用下, M 中的点 (x, y) 被映射到 I 中的 (x', y') 处。用 \hat{m} 表示对点 $m \in M$ 用仿射变换 t 进行变换的结果(即 $\hat{m} = t(m)$)。由于图像中像素点的坐标值为整数,于是用 $[\hat{m}]$ 表示对 \hat{m} 取整后的结果。同理,用 \hat{M} 表示 t 作用于 M 中所有点所产生的二维点集(即 $\hat{M} = t(M)$), $[\hat{M}]$ 表示对 \hat{M} 取整后的结果。于是可定义关于 t 的前向 Hausdorff 距离 $d_{\text{Hausdorff}_F}^{(f_F)}([\hat{M}], I)$ 、后向 Hausdorff 距离 $d_{\text{Hausdorff}_R}^{(f_R)}(I, [\hat{M}])$ 和 Hausdorff 距离 $D_{\text{Hausdorff}_F}^{(f_F, f_R)}([\hat{M}], I)$:

$$D_{\text{Hausdorff}_F}^{(f_F, f_R)}([\hat{M}], I) = \max(d_{\text{Hausdorff}_F}^{(f_F)}([\hat{M}], I), d_{\text{Hausdorff}_R}^{(f_R)}(I, [\hat{M}])) \quad (6)$$

$$d_{\text{Hausdorff}_F}^{(f_F)}([\hat{M}], I) = f_F \text{th} \min_{m \in [\hat{M}]} \|m - i\|$$

$$d_{\text{Hausdorff}_R}^{(f_R)}(I, [\hat{M}]) = f_R \text{th} \min_{i \in I_{\text{box}}} \|i - m\|$$

目标搜索的过程就是搜寻能同时满足 $d_{\text{Hausdorff}_F}^{(f_F)}([\hat{M}], I) \leq \tau_F$ 和 $d_{\text{Hausdorff}_R}^{(f_R)}(I, [\hat{M}]) \leq \tau_R$ 的 t 变换过程。 τ_F 和 τ_R 为两个预先给定的门限值。

3 Rucklidge 快速搜索算法

为了建立有效的搜索机制,Rucklidge 首先对由所有仿射变换 t 所构成的仿射变换空间进行离散化处理。假定所有模板点 $(x, y) = m \in M$ 在模板 M 中的取值范围是 $0 \leq x \leq x_{\max}$ 和 $0 \leq y \leq y_{\max}$ (注意,这里的 x_{\max} 和 y_{\max} 是相对模板 M 的坐标,不同于上一节中的 x_{\max} 和 y_{\max})。经过离散化处理,离散变换空间中的仿射变换同样可以用一个 6 元组来表示。例如,离散空间中的变换 $(i_1, i_2, i_3, i_4, i_5, i_6)$ 对应于连续空间中的变换 $(i_1/x_{\max}, i_2/y_{\max}, i_3/x_{\max}, i_4/y_{\max}, i_5, i_6)$ 。这样处理的好处在于:离散仿射变换空间中的相邻两个变换 t_1 和 t_2 (所谓“相邻”是指 t_1 和 t_2 中 $i_1 \sim i_6$ 等 6 个参数只有一个不同,且相差为 1) 作用于同一点 $(x, y) \in M$ 所得到的两个坐标点 (x'_1, y'_1) 和 (x'_2, y'_2) 在 x 或 y 方向上相差不超过 1 个像素。离散仿射变换空间中的子空间记为 $R, t^{(\text{low})} = [a_{00}^{(\text{low})}, a_{01}^{(\text{low})}, a_{10}^{(\text{low})}, a_{11}^{(\text{low})}, t_x^{(\text{low})}, t_y^{(\text{low})}]$ 表示 R 中参数 $a_{00}^{(\text{low})} \sim t_y^{(\text{low})}$ 为所有变换 $t \in R$ 中的参数 $a_{00} \sim t_y$ 取最小值的变换。同理可定义 $t^{(\text{high})} = [a_{00}^{(\text{high})}, a_{01}^{(\text{high})}, a_{10}^{(\text{high})}, a_{11}^{(\text{high})}, t_x^{(\text{high})}, t_y^{(\text{high})}]$ 表示参数 $a_{00}^{(\text{high})} \sim t_y^{(\text{high})}$ 为所有变换 $t \in R$ 中的参数 $a_{00} \sim t_y$ 取最大值的变换。 R 可由 $t^{(\text{low})}$ 和 $t^{(\text{high})}$ 来唯一确定。简而言之,Rucklidge 算法就是子空间快速分解算法,其具体步骤如下:

(1) 初始化生成包含所有变换 t 的一系列子空间;

(2) 考查当前级的所有子空间 对每个子空间 R ,用 $t^{(\text{low})}$ 对每个模板点 $m \in M$ 进行仿射变换。如果满足 $T_{w,h}^{(\text{distance_box})}([t^{(\text{low})}(m)]) \leq \tau_F$ 的模板点个数不少于 $f_F n_M$,那么就对 R 进行标记。这时 R 中可能包含变换 t ,且满足 $d_{\text{Hausdorff}_F}^{(f_F)}([\hat{M}], I) \leq \tau_F$ 。

(3) 当前级的所有子空间都被考查以后,即可进行子空间的分解,以生成下一级子空间。具体过程是:当前级中每一个被标记过的子空间 R 被分解为 2^6 个大小相等的更小的子空间。重复第(2)步和第(3)步,直到当前级的每个子空间只包含一个仿射变换为止。这时子空间的分解过程停止。

(4) 对于当前级中的每个子空间 t (此时一个子空间即是一个仿射变换),如果同时满足 $d_{\text{Hausdorff}_F}^{(f_F)}([\hat{M}], I) \leq \tau_F$ 和 $d_{\text{Hausdorff}_R}^{(f_R)}(I, [\hat{M}]) \leq \tau_R$,则接受 t ;否则,拒绝 t 。

此算法的特点是经过不同级的子空间分解后,即可快速排除不可能包含满足 $d_{\text{Hausdorff}_F}^{(f_F)}([\hat{M}], I) \leq \tau_F$ 的 t 变换子空间 R 。算法的核心部分是第(2)步。其中的 $T_{w,h}^{(\text{distance_box})}$ 为图像 I 的“盒距离变换”(Box distance transform,是一个二维矩阵),定义为

$$T_{w,h}^{(\text{distance_box})}(x, y) = \min_{\substack{0 \leq x' \leq w \\ 0 \leq y' \leq h}} T_{\text{distance}}(x + x', y + y') \quad (7)$$

式中

$$w = (a_{00}^{(\text{high})} - a_{00}^{(\text{low})}) + (a_{01}^{(\text{high})} - a_{01}^{(\text{low})}) + (t_x^{(\text{high})} - t_x^{(\text{low})});$$

$$h = (a_{10}^{(\text{high})} - a_{10}^{(\text{low})}) + (a_{11}^{(\text{high})} - a_{11}^{(\text{low})}) + (t_y^{(\text{high})} - t_y^{(\text{low})});$$

$T_{\text{distance}}(x, y)$ 为图像 I 的距离变换 (distance transform,是一个二维矩阵),定义为

$$T_{\text{distance}}(x, y) = \min_{i \in I} \|(x, y) - i\| \quad (8)$$

不难证明点 $[\hat{m}](t \in R)$ 在图像 I 中的变化范围被限定在一个以 $[t^{(\text{low})}(m)]$ 为左上角、大小为 $(w+1) \times (h+1)$ 的矩形中。由式(7)可知,如果 $T_{w,h}^{(\text{distance_box})}([t^{(\text{low})}(m)]) > \tau_F$,那么不可能存在这样的 $t \in R$,满足 $T_{\text{distance}}([\hat{m}]) \leq \tau_F$ 。

Rucklidge 算法的不足之处主要有如下两点:

(1) 处于不同级的子空间的 w 和 h 值不相同,所以对每一级都要计算一次 $T_{w,h}^{(\text{distance_box})}$; (2) 即使是在同一级上,对每个不同的模板点 $m \in M$ 都用固定的 w 和 h 值来统计 $T_{w,h}^{(\text{distance_box})}([t^{(\text{low})}(m)])$ 也并不合理,其原因在于:在变换 $t \in R$ 的作用下,不同的 $[\hat{m}]$ 尽管都在图像 I 中的以 $[t^{(\text{low})}(m)]$ 为左上角的矩形中变化,但由于变化范围不尽相同 (即矩形的宽和高不同),而大小为 $(w+1) \times (h+1)$ 的矩形只是这些矩形中最大的一个,因此合理的做法似乎是为每一个不同的模板点 $m \in M$ 都用适合它的 w 和 h 来计算一份 $T_{w,h}^{(\text{distance_box})}$,然而 Rucklidge 指出,这种做法并不可行,因为首先每一份 $T_{w,h}^{(\text{distance_box})}$ 都是一个与图像 I 大小相等的一个二维矩阵;其次,正如第(1)点所指出的,在每一级仍然需要重新计算 $T_{w,h}^{(\text{distance_box})}$,为此,就需要开辟很大的存储空间来保存这些 $T_{w,h}^{(\text{distance_box})}$ 矩阵。作为折衷,他提出了一种对模板 M 进行分区 (zoning) 的办法,其做法是将 M 分成若干大小相等的小区块,每个小区块中的所有模板点共用相同的 w 和 h 值,而不同的小区块则用不同的 w 和 h ,因此分区的数量很关键。如果不对 M 进行划分,那么只能对应仅有一个分区的情况 (即整个模板就是一个大的分区,换言之,就是没有分区,用“Without zoning”表示)。如果对每个模板点都计算一份 $T_{w,h}^{(\text{distance_box})}$,则将对应有很多分区,同时每个分

区只含一个模板点的情况。Rucklidge 采用的做法是将模板分成 4×4 块大小相同的分区(即将整个模板分成 $4 \times 4 = 16$ 个分区,用“ 4×4 Zoning”表示)。

4 改进算法

鉴于采用 $T_{w,h}^{(\text{distance_box})}$ 作为判断准则的不足,因此这里提出一种新的距离变换概念,并称之为“最小正方形距离变换”(Smallest square box distance transform,也是一个二维矩阵)。其定义如下

$$T_{\min_distance_box}(x,y) = \begin{cases} \min_{(w,h) \in Z} \max(w,h) & \text{if } Z \neq \emptyset \\ +\infty & \text{if } Z = \emptyset \end{cases} \quad (9)$$

在式(9)中,集合 $Z = \{(w,h) \mid w \geq 0, h \geq 0, T_{\text{distance}}(x+w, y+h) \leq \tau_F\}$, \emptyset 表示空集。 $T_{\min_distance_box}(x,y)$ 确定了这样一个正方形,它是以下一系列的正方形中最小的一个。这些正方形以 (x,y) 为左上角,并包含至少一个点 (u,v) ,使得 $T_{\text{distance}}(u,v) \leq \tau_F$ 。这个“最小正方形”的边长为 $T_{(\min_distance_box)}(x,y) + 1$ 。

用 $t^{(\text{low})}$ 和 $t^{(\text{high})}$ 分别对模板点 $m = (x,y) \in M$ 进行变换,可以得到两个点 $(x^{(\text{low})}, y^{(\text{low})})$ 和 $(x^{(\text{high})}, y^{(\text{high})})$ (利用离散型仿射变换和连续型仿射变换之间的关系):

$$\begin{aligned} (x^{(\text{low})}, y^{(\text{low})}) &= (a_{00}^{(\text{low})} x/x_{\max} + a_{01}^{(\text{low})} y/y_{\max} + t_x^{(\text{low})}, \\ &\quad a_{10}^{(\text{low})} x/x_{\max} + a_{11}^{(\text{low})} y/y_{\max} + t_y^{(\text{low})}) \\ (x^{(\text{high})}, y^{(\text{high})}) &= (a_{00}^{(\text{high})} x/x_{\max} + a_{01}^{(\text{high})} y/y_{\max} + t_x^{(\text{high})}, \\ &\quad a_{10}^{(\text{high})} x/x_{\max} + a_{11}^{(\text{high})} y/y_{\max} + t_y^{(\text{high})}) \end{aligned} \quad (10)$$

Rucklidge 建议在进行子空间分解时,各个子空间的边长应该彼此相等,即

$$\begin{aligned} a_{00}^{(\text{high})} - a_{00}^{(\text{low})} &= a_{01}^{(\text{high})} - a_{01}^{(\text{low})} = a_{10}^{(\text{high})} - a_{10}^{(\text{low})} \\ &= a_{11}^{(\text{high})} - a_{11}^{(\text{low})} = t_x^{(\text{high})} - t_x^{(\text{low})} = t_y^{(\text{high})} - t_y^{(\text{low})} \end{aligned}$$

这样有利于快速分解。利用这一特点,即可以得到一个关于点 (x,y) 的标量 $W_{x,y}$

$$\begin{aligned} W_{x,y} &= x^{(\text{high})} - x^{(\text{low})} = y^{(\text{high})} - y^{(\text{low})} \\ &= (a_{00}^{(\text{high})} - a_{00}^{(\text{low})})x/x_{\max} + \\ &\quad (a_{01}^{(\text{high})} - a_{01}^{(\text{low})})y/y_{\max} + t_x^{(\text{high})} - t_x^{(\text{low})} \\ &= (a_{10}^{(\text{high})} - a_{10}^{(\text{low})})x/x_{\max} + \\ &\quad (a_{11}^{(\text{high})} - a_{11}^{(\text{low})})y/y_{\max} + t_y^{(\text{high})} - t_y^{(\text{low})} \end{aligned} \quad (11)$$

从公式(11)可知,当变换 t 在子空间 R 中变化时, \hat{m} 的值在图像 I 中一个正方形内变化。这个正方

形以 $(x^{(\text{low})}, y^{(\text{low})})$ 和 $(x^{(\text{high})}, y^{(\text{high})})$ 为其左上角和右下角坐标。注意 \hat{m} 、 $(x^{(\text{low})}, y^{(\text{low})})$ 和 $(x^{(\text{high})}, y^{(\text{high})})$ 的坐标值都是小数。考虑取整操作后的结果,不难证明,当 $W_m \geq T_{\min_distance_box}(\lceil t^{(\text{low})}(m) \rceil)$ 时,可确保 $\lceil t^{(\text{low})}(m) \rceil$ 在一个以 $\lceil t^{(\text{low})}(m) \rceil$ 为左上角的正方形内变化,且此正方形的尺寸不小于前面提到的“最小正方形”,于是从式(9)的定义可知,这时便有可能存在这样的 t ,其满足 $T_{\text{distance}}(\lceil \hat{m} \rceil) \leq \tau_F$ 。这样一来,就可以用以下步骤来代替 Rucklidge 算法中的第(2)步:

考查当前级的所有子空间。对每个子空间 R ,用 $t^{(\text{low})}$ 对每个模板点 $m \in M$ 进行仿射变换。如果满足 $W_m \geq T_{\min_distance_box}(\lceil t^{(\text{low})}(m) \rceil)$ 的模板点个数不少于 f_{FN_M} ,那么就对 R 进行标记。

与 Rucklidge 算法相比,由于本文算法更有效地排除了那些不满足条件 $d_{\text{Hausdorff}_F}^{(f_F)}(\lceil \hat{M} \rceil, I) \leq \tau_F$ 的子空间,因而能够更快地搜索到目标所在位置。同时,改进算法并不影响目标的搜索成功率和定位精度。换言之,在采用相同参数的条件下,Rucklidge 算法如果能搜索到目标,并定位最佳匹配位置,本算法也可得到同样的结果,而且速度更快。

与 $T_{w,h}^{(\text{distance_box})}$ 相比, $T_{\min_distance_box}$ 只需计算一次即可,其结果适用于每一级。此外,从式(11)可以看出, W_m 是根据每个模板点 $m \in M$ 的位置计算得到的,因此,与固定的 w 和 h 相比, W_m 显然更合理。计算 $T_{\min_distance_box}$ 的算法(伪代码)如下

(1) 初始化一个二维整数矩阵 $T_{\min_distance_box}$,其大小与图像 I 相同。 N_{rows} 和 N_{columns} 分别表示 $T_{\min_distance_box}$ 的行数和列数。

(2) for $i=1$ to N_{rows} (outer loop 1)

for $j=1$ to N_{columns} (inner loop 1)

置整数值 $w-h = +\infty$

for $m=i$ to N_{rows} (outer loop 2)

for $n=j$ to N_{columns} (inner loop 2)

if 可找到点 (n,m) 满足 $T_{\text{distance}}(n,m) \leq \tau_F$

then $w-h = \min(\max(m-i, n-j))$,

$w-h$), 跳出 inner loop 2

otherwise do nothing

continue inner loop 2

if $w-h = 0$

then 跳出 outer loop 2

otherwise do nothing

continue outer loop 2

$$T_{\min_distance_box}(j,i)=w-h$$

continue inner loop 1

continue outer loop 1

为了提高效率,还建立了一个查找表用于存储每个模板点 $m \in M$ 的 x/x_{\max} 和 y/y_{\max} 值。这样在使用公式(11)计算 W_m 时更加方便。

5 实验结果

为了验证本文算法的优越性,利用两个例子进行了本文算法与 Rucklidge 算法的对比实验。为了

在相同的条件下进行结果对比,在同一实验中,所有算法都采用相同的参数。所有算法的程序代码全部用 Visual C++ 6.0 编写,并在一台 Dell Precision 340 工作站上实现,操作系统为 Windows 2000 Professional。

实验 1 在这个实验中,是用算法在边缘图像上进行搜索。图像 I 大小为 256×320 像素,模板 M 大小为 72×80 像素,该模板是从另一幅不同角度的图像中剪切而得。其中的待搜索目标为一座纪念物。边缘提取用 Canny 边缘提取算法实现,其他参数取值分别为 $\tau_F = \tau_R = 2$, $f_F = 0.8$, $f_R = 0.9$ 。



(a) 模板(72×80 像素)



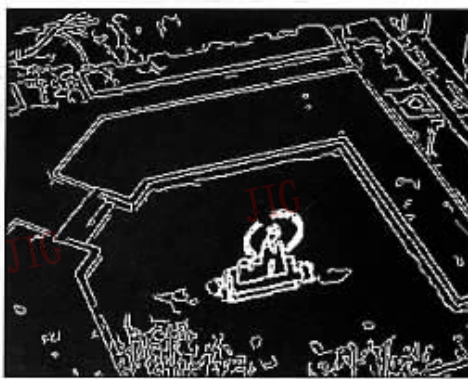
(b) 对图 1(a)进行边缘提取后结果



(c) 搜索图像(256×320 像素)



(d) 对图 1(c)进行边缘提取后结果



(e) 搜索到的模板与图像叠加

图 1 实验 1 目标搜索结果

采用本文算法用时 142.2 s 即可找到最佳的模板匹配位置。而采用文献[3]中的“Without zoning”和“4×4 Zoning”两种方法则分别需耗时 289.1 s 和 245.9 s 才能找到同样的模板匹配位置。

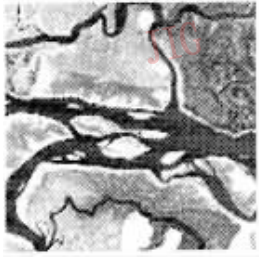
实验 2 为了提高搜索速度,有些目标搜索任务并不是在边缘图像上进行的,而是先采用特征点提取技术提取少量特征点(例如角点等),然后利用这些少量特征点来完成目标搜索,因此在实验 2 中,算法是

在少量特征点图像上实现的。其中搜索图像大小为 128×128 像素,模板大小为 32×80 像素。其他参数取值分别为 $\tau_F = \tau_R = 1.5$, $f_F = 0.8$, $f_R = 0.6$ 。为了简单起见,可通过人工选定少量特征点(这里没有采用特征点提取算法,因为这不是本文研究的重点)。

采用本文算法用时 1.22 s 即可找到最佳的模板匹配位置。而采用“Without zoning”和“4×4 Zoning”两种方法则分别耗时 3.59 s 和 7.55 s 才找



(a) 模板(32×80 像素)

(b) 在图 2(a)中选定的特征点
(用圆圈表示)

(c) 搜索图像(128×128 像素)

(d) 在图 2(c)中选定的特征点
(用十字表示)(e) 搜索到的模板与图像叠加
(特征点叠加)

图 2 实验 2 目标搜索结果

到同样的模板匹配位置。以上两个实验用时见表 1。

表 1 本文方法和 Rucklidge 方法用时的比较 单位:s

用时(s)	“Without zoning”方法	“4×4 Zoning”方法	本文方法
实验 1	289.1	245.9	142.2
实验 2	3.59	7.55	1.22

从表 1 中不难看出,本文方法用时最少,特别是当采用少数特征点进行目标搜索时,这种速度上的优势更加明显。

6 结 论

本文在 Rucklidge 算法的基础上,通过引入“最小正方盒距离变换”这种新的距离变换模型,从而在不牺牲搜索成功率和目标定位精度的情况下,显著地提高了在仿射变换条件下的目标搜索速度。本文方法不但可用于目标搜索,而且通过适当的修改也可以用于其他计算机视觉和模式识别任务,例如图像配准和目标姿态判别等。

本文方法还可以同其他方法,例如并行计算技术^[3]和分级匹配技术^[4]结合起来,以满足某些实时任务的需要。

参 考 文 献

- 1 Huttenlocher D P, Klanderma G A, Rucklidge W J. Comparing images using the Hausdorff distance[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1993, 15(9): 850~863.
- 2 Huttenlocher D P, Rucklidge W J. A multi-resolution technique for comparing images using the Hausdorff distance [R]. Technical Report 1321, Cornell University, Department of Computer Science. Ithaca, New York State USA, 1992.
- 3 Rucklidge W J. Efficiently locating objects using the Hausdorff distance[J]. International Journal of Computer Vision, 1997, 24(3):251~270.
- 4 Kwon O K, Sim D G, Park R H. Nonparametric hierarchical

Hausdorff distance matching algorithm[J]. Optical Engineering, 2000, 39(7):1917~1927.



彭晓明 1974 年生,2000 年获中国科学院光电技术研究所工学硕士学位,现为华中科技大学图象识别与人工智能研究所博士研究生。主要研究方向为图像处理、图像融合和目标识别等。



丁明跃 1961 年生,1988 年获华中理工大学工学博士学位,博士生导师,教授,华中科技大学图象识别与人工智能研究所副所长。主要研究方向为图像处理、图像融合和目标识别等。已发表学术论文 80 多篇。



周成平 1957 年生,1979 年毕业于西安交通大学,硕士生导师,副教授。主要研究方向为图像处理、图像融合和目标识别等。已发表学术论文 20 多篇。



张天序 1947 年生,1989 年获浙江大学博士学位,博士生导师,教授,华中科技大学图象识别与人工智能研究所所长。主要研究方向包括精制导智能化信息处理、计算机视觉等。已发表学术论文 100 多篇。