

对 Freeman 链码分析的角点检测算法^①

刘相湖, 王 涛, 张小哲

(华南师范大学 计算机学院, 广州 510631)

通讯作者: 刘相湖, E-mail: liuxianghu_93@163.com

摘 要: 针对传统的基于轮廓曲线的角点检测算法需要计算曲率和选取阈值的不足, 提出一种对 Freeman 链码分析的角点检测算法, 首先通过图像边缘检测, 轮廓提取得到轮廓的 Freeman 链码, 当链码发生变化时分析其连续前后多个点的链码是否符合一定的规则来判定角点, 无需经过传统的角点阈值选取, 曲率计算等步骤. 实验通过与 He&Yung、CPDA、Fast-CPDA 和 ARCSS 角点检测器比较, 结果表明本文算法在角点检测时准确率 (ACU) 最高; 在变换实验中, 本算法的平均重复率 (AR) 最高, 由此可以得出本算法具有良好的角点检测性能.

关键词: 角点检测; Freeman 链码; 轮廓提取; 链码统计; 链码分析

引用格式: 刘相湖, 王涛, 张小哲. 对 Freeman 链码分析的角点检测算法. 计算机系统应用, 2018, 27(4): 202-208. <http://www.c-s-a.org.cn/1003-3254/6314.html>

Corner Detection Algorithm Based on Freeman Code Analysis

LIU Xiang-Hu, WANG Tao, ZHANG Xiao-Zhe

(School of Computer, South China Normal University, Guangzhou 510631, China)

Abstract: In view of the traditional need to calculate the curvature and the insufficient threshold selection of detection algorithm angular profile curve, this study proposes an angle detection analytic algorithm based on Freeman code by first edge detection, contour extraction to give Freeman code profile. When the chain code changes, it analyzes whether the chain code of a number of points before and after it is consistent with certain rules to determine the corner point, without the need of traditional corner threshold selection, curvature calculation and other steps. Compared with He & Yung, CPDA, Fast-CPDA, and ARCSS corner detector, the experimental results show that the proposed algorithm has the highest accuracy (ACU) in detecting the corner. The transformation experiments show that the average repetition (AR) is the highest, which shows that the proposed algorithm has good corner detection performance.

Key words: corner detection; Freeman code; neighbor factor; chain code statistics; chain code analysis

图像中的角点是图像的重要特征, 具有旋转不变性, 决定了图像形状, 可以降低图像信息的存储效率, 在目标跟踪, 目标检测, 图像匹配, 图像轮廓拟合等领域都有重要的应用价值. 近几十年来, 国内外学者提出的图像角点检测算法^[1], 各有各的优缺点, 大致可分为三大类: 基于灰度强度的角点检测、基于模型的角点检测、基于边缘轮廓的角点检测^[1]. 基于灰度强度的方法是通过图像的一阶或二阶导数查找, 经典算法代表有

Harris 算法^[2], Harris 算法通过微分运算和自相关矩阵来检测角点, 稳定性高, 但是高斯平滑函数的窗口可控性差, 定位精度较差^[3]. 基于模型的方法是通过将一小块图像与预定义的模型相匹配来查找角点, SUSAN 算法^[4]就是这类角点检测的代表, SUSAN 通过一个圆形模板实现角点检测, 通过模板中所有与圆心点像素相似的点组成的区域大小判断角点, 该算法不对图像求导, 精度较好, 具有一定的抗噪性, 但是相似函数计算较复杂, 且

① 收稿时间: 2017-08-08; 修改时间: 2017-08-28; 采用时间: 2017-09-06; csa 在线出版时间: 2018-03-31

需要人工设定阈值^[4]. 基于边缘轮廓的方法主要是通过分析图像边缘形状来检测角点的, 包括五个步骤^[1]: 边缘提取和选择、曲线平滑、曲率估计、检测角点和角点跟踪筛选出最终的角点. 近年来, 基于边缘轮廓方法吸引了广泛的关注, 许多此类优秀的角点检测算法被提出^[5]. 1998年 Mokhtarian 等人提出的著名 CSS 角点检测算法^[4], 利用 Canny 边缘检测算子提取图像边缘, 从边缘中提取边缘轮廓, 并填充边缘轮廓缺口, 在大尺度下得到轮廓曲率极大值点, 通过阈值选定候选角点, 大尺度到小尺度下对候选角点重新定位, 比较 T-角点与曲率极大值检测的角点, 剔除相距相近的点, 该算法获得良好的角点检测效果. 但 CSS 有两个主要的问题^[6], 一是曲率估计对对轮廓局部变化和噪声较为敏感, 二是很难选择合适的高斯尺度平滑边缘轮廓. 在 CSS 基础上, 通过考虑检测中局部曲率, He 和 Yung^[7]提出了一种使用自适应曲率阈值和动态支撑区域的检测器. 随后 Awrangjeb 和 Lu 利用点到弦的距离累加技术提出了一种稳健的角点检测算法——CPDA^[8], CPDA 具有较高的可重复性和较低的定位误差. 之后又提出了 CPDA 的快速算法^[9]. 但 CPDA 有一个缺点是估计拐角的曲率值与角度可能不成正比^[5]. 还有一些角点检测算法, 利用 Freeman 链码多边形近似方法查找角点, 两边的交点判定为角点^[1], 文献[10]对疑似角点统计链码做进一步筛选角点, 但文献中的方法先使用的是 CSS 角点检测算法, 仍有曲率计算和角点阈值选取问题, 并且时间复杂度较高; 文献[11]通过判断待定角点前后三个点的 Freeman 链码是否相同进行检测, 该方法造成错误角点较多, 且文中方法过于繁琐.

本文在文献[10,11]的启发下, 优化文献[11]的链码分析方法, 同时引入文献[10]的链码统计思路, 提出一种新的角点检测算法: 提取图像边缘轮廓 Freeman 链码, 分析链码发生变化时连续多个点的码值是否符合一定的规则来判定角点. 该算法将链码分析和链码统计相结合, 不仅使得算法变得更加简便, 而且避免了传统方法中轮廓曲率计算和阈值选取, 通过实验结果得出本文算法对于图像角点检测结果准确性高, 具有良好稳健性.

1 Freeman 链码生成

1.1 图像预处理

本文图像在预处理方面与文献[5]一致采用 Canny

算法^[12]检测边缘, Canny 算法抑制了多响应边缘, 提高边缘的定位精度, 具有一定的抗噪能力. Canny 算法检测边缘的步骤如图 1.

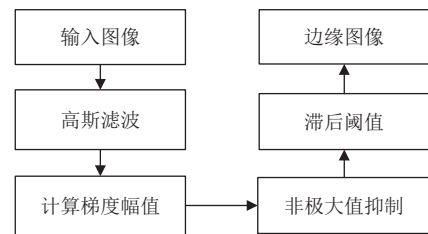


图 1 Canny 边缘检测步骤

1.2 Freeman 链码

链码是通过带有给定方向的单位长度的线段序列来描述轮廓的边界, 链码的表示方法有基于 4-邻接和 8-邻接^[12], 如图 2(a) 和图 2(b) 所示, 8-邻接链码比 4-邻接链码增加了 4 个斜方向, 而较多的基于 Freeman 链码的算法^[10-13]都采用 8-邻接来表示链码. 在实际应用中, 边界跟踪法^[12]产生轮廓的链码, 4-邻接检测边界只需要搜索 4 个方向, 8-邻接则需要搜索 8 个方向, 我们实验所用图都是点阵图, 当轮廓是一条直线时, 4-邻接链码要比 8-邻接链码要快捷的多; 当轮廓是一条斜边时, 用 4-邻接链码只能搜索上下左右 4 个方向, 表示轮廓时会丢失大量的边缘信息, 而 8-邻接链码正好与像素点的实际情况相符, 能够准确地描述中心像素点与其邻接点的信息, 能较好地保留轮廓边缘信息. 因为任意一个像素周围均有 8 个邻接点, 因此本文算法采用的是 8-邻接链码.

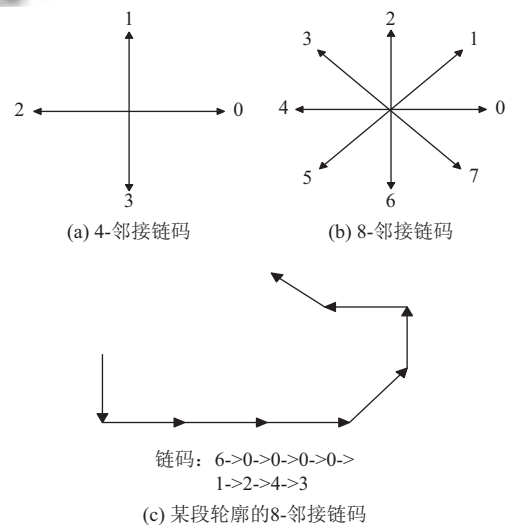


图 2 链码示意图

一条轮廓曲线 Freeman 链码可以定义为: $\{a_{i-1} \rightarrow a_i\}^n$, 其中 $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7\}$, n 表示轮廓点集数. 如图 2(c), 是一段轮廓的 8-邻接链码.

1.3 轮廓跟踪得到 Freeman 链码

基于八邻域边界跟踪可以得到轮廓的 Freeman 链码, 轮廓跟踪^[13]方法如算法 1.

算法 1. 轮廓跟踪生成 Freeman 链码算法.

- 1) 分轮廓搜索, 找到图像最左上角的一个边界轮廓点 P_0 作为搜索起点, 并以链码值 $dir=0$ 为最开始搜索方向;
- 2) 按八邻域的逆时针方向搜索起点的下一个点, 每搜索一次链码值方向逆时针旋转 45° , 即 $dir+1$; 如果搜索到新的边界点 P_i , 将该链码值赋值给它前一个点即点 P_{i-1} 的方向, 再以此点为八邻域的中心点, 链码值方向顺时针旋转 90° , 即 $dir-2$ 作为该点的开始搜索方向, 继续搜索;
- 3) 重复 2) 搜索方法, 当搜索到起始点 P_0 时, 整个轮廓搜索完毕.

通过上述轮廓跟踪算法就得到了边缘轮廓的 Freeman 链码, 我们假定逆时针为正方向, 分轮廓将轮廓点坐标和链码值存储, 存储 FMP 格式有如下的定义:

```
struct FMP
{
    int x; //点的图像 x 坐标
    int y; //点的图像 y 坐标
    int a; //下一个点的方向
}
```

2 对 Freeman 链码方向分析进行角点检测

文献[11]通过对真实角点周围的几个点的 Freeman 链码分析, 文中分析出要想成为一个真实角点, 分情况讨论了前后连续的三个点的链码值是否相同, 相同则为角点, 不相同则不是角点, 但这样检测出角点误检漏检点较多, 且分析方法较为繁琐; 而文献[10]在改进 CSS 角点检测之后利用链码统计思路, 统计疑似角点周围几个点的链码值相同的点数再做进一步的判断, 较好地排除了一些错误角点, 但算法时间复杂度较高, 依然存在曲率计算和阈值选取问题. 本文算法在链码分析上做了较大的改善, 同时结合了链码统计思路, 提出了链码分析规则 1. 如图 3 所示, A 点有可能是角点, A 的链码值为 7, 此时有 4 种情况: 如果 B 点之后的链码序列是 {00000}, 那么 A 点就不是角点, 视为一条斜线的一点; 如果 B 点之后的序列是 {77777}, 那么 A 点视为两条夹角为 135° 的直线的交点, 如果 B 点之后的序列是 {66666}, 那么 A 点也视为两条夹角为 90° 的直

线的交点; 如果 B 点之后的序列是 {55555}, 那么 A 点也视为两条夹角为 45° 的直线的交点, 后 3 种情况都是角点, 这里的 4 种情况包含了文献[11]中提及的所有情况的分析.

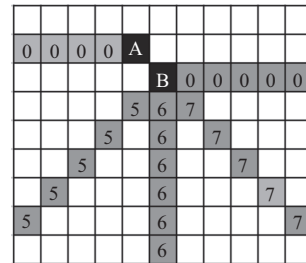


图 3 疑似角点周围点的方向

由于 A 点的链码值为 7, 角点符合上述 3 种情况的链码序列值分别为 7, 6, 5. 综上所述, 本文提出下述规则.

规则 1. 如果某一个真实角点存在, 那么它的前后几个点的链码方向与它的链码方向夹角都是小于或者等于 90° .

2.1 算法基本思路

本文算法角点检测思路如下:

- (1) Canny 算法进行图像边缘检测;
- (2) 轮廓跟踪法得到轮廓的 Freeman 链码;
- (3) 根据图像轮廓点的链码, 求出该轮廓所有点的链码差 d_i ^[11];
- (4) 对于凸起或者凹陷处轮廓点进行链码修复;
- (5) 分析链码发生变化的点的周围多个链码: (a) 对于 $d_i = \pm 3$ 的轮廓点判定为角点; (b) 对于 $d_i = \pm 1$ 或 ± 2 的轮廓点判断附近多个点的链码值是否符合上述规则 1.

2.2 算法具体描述

2.2.1 计算轮廓链码差

轮廓链码差^[11]即轮廓曲线上的点和它的前一个点的链码方向之差, 定义数组 d_i , 存储两个点的链码差, 令 $f(i) = a_i - a_{i-1}$, 则整个轮廓的 d_i 数学计算公式定义:

$$d_i = \begin{cases} f(i), & \text{if } (|f(i)| \leq 4) \\ f(i) + 8, & \text{if } (|f(i)| > 4 \&\& f(i) \leq 0) \\ f(i) - 8, & \text{if } (|f(i)| > 4 \&\& f(i) > 0) \end{cases} \quad (1)$$

其中 d_i 为正表示正方向 (逆时针), 为负表示反方向 (顺时针).

计算得到的 d_i 表示前后两点轮廓之间的关系^[13]:

- (1) $d_i = \pm 1$ 表示前后两个方向夹角为锐角 45° ;

- (2) $d_i = \pm 2$ 表示夹角为直角 90° ;
 (3) $d_i = \pm 3$ 表示夹角为钝角 135° ;
 (4) $d_i = \pm 4$ 表示夹角为直线 180° ;
 (5) $d_i = 0$ 表示前后两个方向夹角 0° , 不发生变化,

将不做下面的讨论范围。

2.2.2 凸起点链码修复

在轮廓曲线中, 由于噪声使得在轮廓查找中或多或少有些像素点出现误差。例如原本一段直线中某像素点突然的凸起或者凹陷, 会造成角点检测错误或漏检, 需要将链码进行修复。如图4, 为了便于计算, 这里只是进行链码简单修复。



图4 轮廓的凸起点

处理过程: 判断变化点是否为凸起点, 若为凸起点, 则进行链码修复, 并修改对应的 d_i 值。对每一条轮廓循环查找。

为了修复凸起点的链码, 设计了算法2。

算法2. 凸起点链码修复的算法。

- 1) 输入所有轮廓 FMP, 轮廓索引号 $index$;
- 2) 循环判断 d_i 是否等于 ± 2 , 如果不等于, 则判断下一个 d_i ;
- 3) 如果 d_i 等于 ± 2 , 需要进一步判断 $a(i-2)$ 是否等于 $a(i+1)$, 如果相等, 则将 $a(i-2)$ 的值赋值给 $a(i)$ 和 $a(i-1)$, 同时 $d_{(i-1)}$ 和 d_i 都赋值为 0。
- 4) 当所有点都修复完, 则循环结束。

2.2.3 根据链码差分析出角点

循环判断 d_i 是否等于0, 如果不等于0, 说明前后两点的链码值发生了变化, 初步认定为疑似角点, 根据1.2可知, 我们这里把点 $i-1$ 的链码到点 i 的链码变化定义为: $a_{i-1} \rightarrow a_i$ 。对疑似角点的前后相邻多个点链码值进行分析, 设置相邻因子 NF , 同时为了弥补图像的边缘丢失和满足一定的曲线的曲率变化, 设置相同点数 (链码与变化前后相同的点数) S , 输出的角点存储到 $vecCorner$ 中。

对所有轮廓 FMP 的每一条轮廓 (FMP[$index$], $size > 20$, $index$ 表示轮廓索引号) 循环查找角点。对轮廓 FMP[$index$] 的链码差计算和链码修复完成后, 再重新对该轮廓循环查找角点, 对链码差分 d_i 分以下3种情况

讨论, 设置标记数 $Q12=0$:

- (1) 若 $d_i = \pm 3$; 则该点 P_i 被判定为角点, $Q12=NF$;
- (2) 若 $d_i = \pm 2$; 循环观察该点 P_i 轮廓前 NF 个点和点 P_i 轮廓后 NF 个点的链码, 如图6, 即 $pre_dirs = \{a_{i-2}, a_{i-3}, \dots, a_{i-NF-1}\}$ 和 $dirs = \{a_{i+1}, a_{i+2}, \dots, a_{i+NF}\}$, 计算 pre_dirs 与 a_i 相同的个数和 $dirs$ 与 a_{i-1} 相同的个数之和 num , 若 $num \leq S$, $Q12=NF$;

(3) 若 $d_i = \pm 1$; 对 $d_i = 1$ 和 $d_i = -1$ 分析。分别计算与 a_{i-1} 和 a_i 的链码方向的夹角小于等于 90° 的链码集合。

1) 当 $d_i = 1$, 即链码变化前后为逆时针旋转。如下计算方法得到的值存入数组 $K1$ 和 $K2$ 中:

计算 a_{i-1} 的顺时针方向的两个链码组成数组 $K1$, 即 $K1 = \{a_{i-1}, a_{i-1}-1, a_{i-1}-2\}$;

计算 a_i 的逆时针方向的两个链码组成数组 $K2$, 即 $K2 = \{a_i, a_i+1, a_i+2\}$ 。

例如 $0 \rightarrow 1$ 逆时针方向变化: 计算 0 的顺时针方向的两个链码分别为 7 和 6, 组成数组 $K1 = \{0, 7, 6\}$; 计算 1 的逆时针方向的两个链码分别为 2 和 3, 组成数组 $K2 = \{1, 2, 3\}$, 如图5(a)。

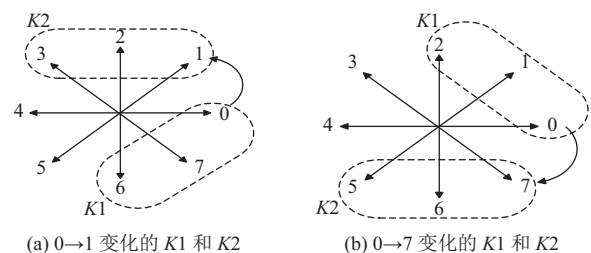


图5 方向分析示意图

2) 当 $d_i = -1$, 即链码变化前后为顺时针旋转。如下计算方法得到的值存入数组 $K1$ 和 $K2$ 中:

计算 a_{i-1} 的逆时针方向的两个链码组成数组 $K1$, 即 $K1 = \{a_{i-1}+1, a_{i-1}+1, a_{i-1}+2\}$;

计算 a_i 的顺时针方向的两个链码组成数组 $K2$, 即 $K2 = \{a_i, a_i-1, a_i-2\}$ 。

例如 $0 \rightarrow 7$ 顺时针方向变化: 计算 0 的逆时针方向的两个链码分别为 1 和 2, 组成数组 $K1 = \{0, 1, 2\}$; 计算 0 的顺时针方向的两个链码分别为 7 和 6, 组成数组 $K2 = \{0, 7, 6\}$, 如图5(b)。

上述得到 $K1$ 和 $K2$ 数组后, 再进行如下运算: 对于该点 P_i 的轮廓前面 NF 个点的链码, 如图6, 即 $pre_dirs = \{a_{i-2}, a_{i-3}, \dots, a_{i-NF-1}\}$ 和该点 P_i 的轮廓后面 NF 个点的链码, 如图6, 即 $dirs = \{a_{i+1}, a_{i+2}, \dots, a_{i+NF}\}$,

计算 pre_dirs 在数组 $K1$ 中存在相同链码的个数 N_1 和 $dirs$ 在数组 $K2$ 中存在相同链码的个数 N_2 , 满足 $N_1 + N_2 \geq 2NF - S$, 则 $Q12=NF$.

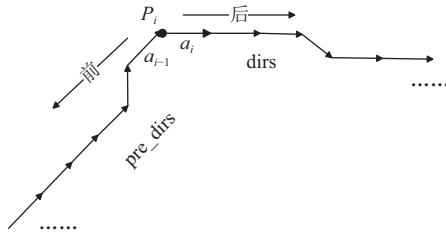


图6 疑似角点 P_i 轮廓前后链码示意图

选择执行完上述3种情况后, 角点的判定条件是: $Q12=NF$, 若满足此条件的点再与 $vecCorner[index]$ 里的角点比较轮廓差, 若相差小于 NF , 则与它最近的角点位置取中间位置作为新角点, 且该点只能比较一次; 若相差大于等于 NF , 则直接判定为角点, 存入 $vecCorner[index]$.

3.3 算法复杂度分析

本文算法在角点检测部分主要是分情况分析链码方向, 对疑似角点的周围几个点的链码值进行分析比较, 不涉及乘除法运算和曲率运算, 算法角点检测部分计算量较少, 计算过程是分轮廓计算, 得到的角点也是分轮廓存储. 本文算法的角点检测部分(不包括轮廓跟踪部分)的时间复杂度为 $O(n)$, 空间复杂度为 $O(n)$, 其中 n 为所有轮廓的总点数.

3 实验结果分析

为了验证本文算法的有效性和准确性, 实验在 PC(Inter(R) Core(TM) i5-3450 CPU @ 3.10GHz, 8G 内存) 机上利用 MATLAB R2014b 进行实验, 与基于边缘轮廓法影响较广且较新的一些算法: ARCSS^[14], CPDA^[8], Fast-CPDA^[9] 和 He&Yung^[7] 做比较. 实验所用图和参考图均来自文献[15-18], 如图7所示为角点参考图, 其中两幅二值图, 两幅实物灰度图.

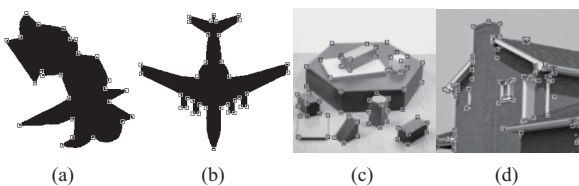


图7 本文实验所用图

3.1 准确性实验

本文先以精确度 ACU 评估五种角点检测算法的准确性^[17]. 假设 N_0 是原始图像中的真实角点, N_a 是算法检测出的正确角点数, N_f 是算法从原始图像检测出的角点数. 另外, 如果检测到的最小距离小于或者等于所设定的距离阈值(本文设置为4, 即4个像素的距离), 则判断该角点检测正确^[19]. 精确度 ACU 定义如下:

$$ACU = \frac{N_a}{2} \left(\frac{1}{N_0} + \frac{1}{N_f} \right) \quad (2)$$

为了对比较的公平性, 在比较过程中, 五种算法在提取边缘轮廓选取相同参数, 均采用 Canny 算法, 低阈值 $L=0.2$, 高阈值 $H=0.35$, 检测时选取算法检测最优结果时的参数^[1]进行实验比较. 本文算法参数设置: 相邻因子 $NF=6$, 相同点 $S=2$. 对飞机检测结果如图8, 具体结果如下表1.

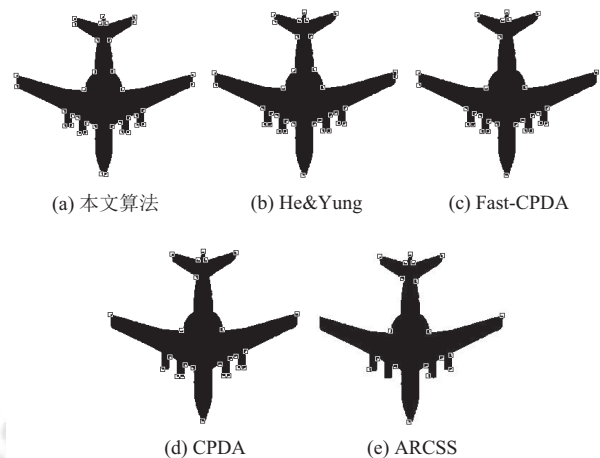


图8 各算法对飞机图像的角点检测

表1 准确性实验结果

算法	正确角点	检测角点	丢失角点与 错误角点之和	平均ACU (%)
ARCSS	81	83	116	71.34
CPDA	109	111	88	79.40
Fast-CPDA	121	123	76	82.57
He&Yung	167	194	55	88.37
本文算法	158	171	50	89.66

由图8和表1可以看出, ARCSS、CPDA 和 Fast-CPDA 检测到的错误角点较少, 但是丢失角点都较多; 虽然 He&Yung 检测到的角点最多, 但是错误角点也是最多的, 而本文算法检测到的丢失角点与错误角点之和最少, 四幅图的平均准确率也是最高, 可以看出本文

算法具有良好的角点检测准确性。

3.2 稳定性实验

为了验证算法在图像变换下的鲁棒性,实验利用图7的四幅基础图建立一个数据集,经过如下变换^[15]:

(1) 旋转变换: 旋转角度区间为 $[-90^\circ, +90^\circ]$, 每 10° 旋转一次, 除 0° 之外;

(2) 等比例尺度变换: 尺度因子变化区间为 $[0.5, 2]$, 以0.1逐步增加, 除1.0之外;

(3) 非等比例尺度变换: x 的变换范围为 $[0.7, 1.5]$, y 的变换范围为 $[0.5, 1.8]$, 以0.1逐步增加, 除了都为1.0;

(4) 组合变换: 角度旋转区间为 $[-30^\circ, +30^\circ]$, 每 10° 旋转一次, 除 0° 外; 图像横纵坐标 x, y 变换范围: $[0.8, 1.2]$, 以0.1为单位逐步增加, 其中 x, y 都为1.0除外;

(5) 添加噪声: 噪声范围 $[0.005, 0.05]$, 以0.005为单位逐步添加零均值高斯白噪声。

图像变换实验时, 以平均重复率 (AR) 和定位误差 (LE) 来评价角点检测算法^[15]。假设 N_0 是算法检测初始图像的角点个数, N_T 是算法检测变换后图像的角点个数, N_r 是两者重复的角点个数。则平均重复率为:

$$R_a = \frac{N_r}{2} \left(\frac{1}{N_0} + \frac{1}{N_T} \right) \quad (3)$$

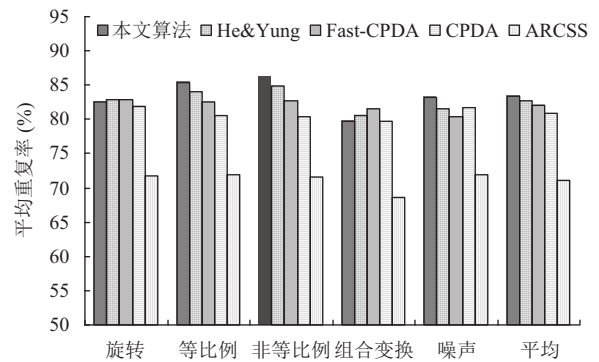
平均重复率越高, 说明算法在越稳定。定位误差为:

$$L_e = \sqrt{\frac{1}{N_r} \sum_{i=1}^{N_r} [(x_{ii} - x_{0i})^2 + (y_{ii} - y_{0i})^2]} \quad (4)$$

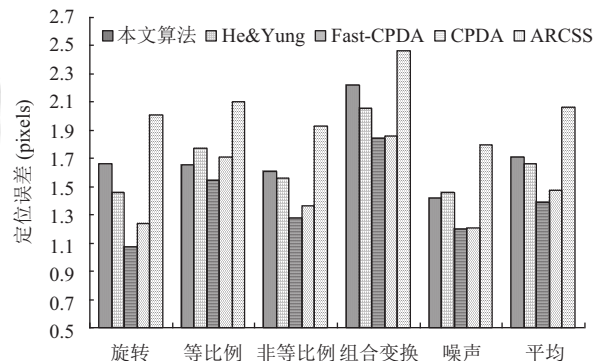
其中, (x_{0i}, y_{0i}) 和 (x_{ii}, y_{ii}) 分别为初始图像和变换后图像位置的第 i 个重复的角点。这里, 如果两点相差的距离不大于4个像素, 则认为它们是重复的^[19]。如图9是上述5种变换下各算法的AR和LE。

从图9可以看出, 本文算法在除了旋转变换和组合变换下平均重复率不及He&Yung和Fast-CPDA, 其他变换下平均重复率都要高于其他算法; 不过定位误差要高于CPDA、Fast-CPDA和He&Yung算法。最终的平均值如表2。

由表2可以看出, 本文算法在定位误差方面虽然没有CPDA、Fast-CPDA和He&Yung算法好, 但要好于ARCSS, 并且本文算法在5个检测算法具有最高的平均重复率, 说明本文算法具有良好的角点检测稳健性。



(a) 5种变换下各算法的平均重复率



(b) 5种变换下各算法的定位误差

图9 5种变换下各算法的AR和LE

表2 稳定性实验比较结果

序号	AR (%)		LE (pixels)	
1	本文算法	83.41	Fast-CPDA	1.39
2	He&Yung	82.76	CPDA	1.48
3	Fast-CPDA	81.99	He&Yung	1.67
4	CPDA	80.86	本文算法	1.71
5	ARCSS	71.14	ARCSS	2.05

此外, 本文提出的相邻因子 NF 和相同点 S 在图像的等比例尺度变换时可以与尺度因子等比例增减。实验过程中, 为了实验对比的公平性, 本文没有采用等比例 NF 和 S , 都以不改变 NF 和 S 进行角点检测实验。在实际应用中, 图像执行等比例尺度变换时, 可以采用等比例增减 NF 和 S 进行角点检测, 效果更佳。

3.3 算法运行时间实验

我们记录上述五种算法运行图7中4幅图所耗时间, 分别运行10次, 然后计算平均运行时间 (除去统一使用的Canny算法运行时间)。运行时间如表3

表3 五种角点检测算法运行时间 (单位: s)

算法	He&Yung	Fast-CPDA	CPDA	本文算法	ARCSS
运行时间	0.0723	0.0831	0.0996	0.1044	0.1053

由表3可以得出, He&Yung 运行时间最快, 本文算法运行时间稍长于 Fast-CPDA、He&Yung 和 CPDA, 主要在轮廓跟踪得到链码部分耗时较多。

通过以上3种实验对比分析, He&Yung 运行时间最快, Fast-CPDA 算法定位误差最低, 但本文算法的角点准确率和平均重复率都是最高的, 且本文算法不需要阈值选取。

4 结束语

本文提出一种基于 Freeman 链码方向分析的角点检测算法, 通过巧妙地分析链码方向, 计算链码发生变化时周围多个链码的值是否在一定的数组中来判定角点, 而不需要经过传统的曲线曲率计算检测角点, 同时避免了阈值选取的问题, 实验结果得出本文算法在角点检测时具有良好的检测准确性和稳健性, 角点存储分轮廓具有顺序性, 算法很好理解, 角点检测部分时间复杂度小。不过在算法前期需要链码的提取, 使得算法运行时间要比 He&Yung、Fast-CPDA、CPDA 算法稍长, 另外, 本文算法在组合变换中检测角点不够理想, 在以后工作中会加以改进。

参考文献

- 1 Awrangjeb M, Lu GJ, Fraser CS. Performance comparisons of contour-based corner detectors. *IEEE Transactions on Image Processing*, 2012, 21(9): 4167–4179. [doi: 10.1109/TIP.2012.2200493]
- 2 Harris C, Stephens M. A combined corner and edge detector. *Proceedings of the 4th Alvey Vision Conference*. Manchester, England. 1988. 147–151.
- 3 洪改艳, 芮廷先, 俞伟广, 等. Harris 角点检测的优化算法. *计算机系统应用*, 2017, 26(4): 169–172.
- 4 章为川, 孔祥楠, 宋文. 图像的角点检测研究综述. *电子学报*, 2015, 43(11): 2315–2321. [doi: 10.3969/j.issn.0372-2112.2015.11.026]
- 5 Zhang WC, Shui PL. Contour-based corner detection via angle difference of principal directions of anisotropic Gaussian directional derivatives. *Pattern Recognition*, 2015, 48(9): 2785–2797. [doi: 10.1016/j.patcog.2015.03.021]
- 6 李云红, 姚韵, 贾利娜. 基于图像轮廓的角点检测算法. *计算机与数字工程*, 2016, 44(10): 2015–2019. [doi: 10.3969/j.issn.1672-9722.2016.10.032]
- 7 He XC, Yung NHC. Corner detector based on global and local curvature properties. *Optical Engineering*, 2008, 47(5): 057008. [doi: 10.1117/1.2931681]
- 8 Awrangjeb M, Lu GJ. Robust image corner detection based on the chord-to-point distance accumulation technique. *IEEE Transactions on Multimedia*, 2008, 10(6): 1059–1072. [doi: 10.1109/TMM.2008.2001384]
- 9 Awrangjeb M, Lu GJ, Fraser CS, *et al.* A fast corner detector based on the chord-to-point distance accumulation technique. *Proceedings of 2009 Digital Image Computing: Techniques and Applications*. Melbourne, VIC, Australia. 2009. 519–525.
- 10 曾接贤, 李炜焯. 曲率尺度空间与链码方向统计的角点检测. *中国图象图形学报*, 2014, 19(2): 234–242.
- 11 黄琼丹. 一种基于轮廓分析的图像特征点检测方法. *微计算机信息*, 2009, 25(25): 123–125. [doi: 10.3969/j.issn.1008-0570.2009.25.051]
- 12 Sonka M, Hlavac V, Boyle R. *图像处理、分析与机器视觉*. 兴军亮 艾海舟, 译. 4版. 北京: 清华大学出版社, 2016. 97–106.
- 13 吴桐树, 张瑞林, 邹敏. 基于 Freeman 链码的 B 样条曲线轮廓拟合. *计算机系统应用*, 2014, 23(8): 130–134.
- 14 Awrangjeb M, Lu GJ, Murshed M. An affine resilient curvature scale-space corner detector. *Proceedings of 2007 IEEE International Conference on Acoustics, Speech and Signal Processing*. Honolulu, HI, USA. 2007. I-1233–I-1236.
- 15 Zhang SZ, Yang D, Huang S, *et al.* Robust corner detection using the eigenvector-based angle estimator. *Journal of Visual Communication and Image Representation*, 2017, 45: 181–193. [doi: 10.1016/j.jvcir.2017.01.020]
- 16 Xing YX, Zhang DY, Zhao JH, *et al.* Robust fast corner detector based on filled circle and outer ring mask. *IET Image Processing*, 2016, 10(4): 314–324. [doi: 10.1049/iet-ivr.2014.0952]
- 17 徐玲. 基于图像轮廓的角点检测方法研究[博士学位论文]. 重庆: 重庆大学, 2009.
- 18 李冬晴. 图像角点检测算法与测评技术研究[硕士学位论文]. 苏州: 苏州大学, 2015.
- 19 赵亚利, 章为川, 李云红. 图像边缘轮廓自适应阈值的角点检测算法. *中国图象图形学报*, 2016, 21(11): 1502–1514. [doi: 10.11834/jig.20161110]