

# 基于改进粒子群算法的彩色图像边缘检测方法

刘笃晋<sup>a</sup>, 孙淑霞<sup>b</sup>, 丁照宇<sup>b</sup>, 李思明<sup>b</sup>

(成都理工大学 a. 信息科学与技术学院; b. 网络教育学院, 成都 610059)

**摘 要:** 针对传统图像边缘检测方法检测效果不理想的问题, 利用四元数的矢量旋转原理, 将一种改进后的粒子群优化算法引入四元数图像边缘检测中, 提出一种新的彩色图像边缘检测方法。实验结果表明, 该方法对彩色图像的边缘检测效果较好, 能够提取图像纹理细节, 且算法稳定、容易收敛, 边缘检测速度也较快。

**关键词:** 彩色图像; 四元数; 四元数矢量旋转; 粒子群优化; 边缘检测

## Color Image Edge Detection Method Based on Improved Particle Swarm Algorithm

LIU Du-jin<sup>a</sup>, SUN Shu-xia<sup>b</sup>, DING Zhao-yu<sup>b</sup>, LI Si-ming<sup>b</sup>

(a. College of Information Science and Technology; b. College of Network Education, Chengdu University of Technology, Chengdu 610059, China)

**【Abstract】** Because of the traditional edge detection inadequacy, quaternion's vector rotating theory is used, improved particle swarm algorithm is introduced to quaternion edge detection. A new color image edge detection method is presented. Experimental results show that the proposed method of color image edge detection has good effect. It can extract the image detail, and the algorithm is stable, easy to convergence, edge detection a little faster.

**【Key words】** color image; quaternion; quaternion vector rotation; Particle Swarm Optimization(PSO); edge detection

DOI: 10.3969/j.issn.1000-3428.2011.15.060

### 1 概述

边缘检测<sup>[1]</sup>是图像处理中最基本的问题之一, 是后续图像分割、特征提取等的基础。基于灰度图像的边缘检测<sup>[2]</sup>方法已经相当成熟, 近年来, 数字彩色图像的广泛运用使彩色数字图像处理技术日益受到关注, 其常用的彩色表示方法是 RGB 三色模型空间, 而传统的边缘检测算法如 Robert 算子、Sobel 算子、Prewitt 算子、Canny 算子和 Laplacian 算子等, 对于彩色图像, 基本上都是灰度图像的直接扩展, 或者是对彩色分量分别进行检测再以某种方法将结果合成, 没有将一个彩色图像像素作为一个整体来处理, 并且检测时没有考虑单个像素和图像全部像素的关系, 因而边缘检测效果不理想。

四元数于 1843 年被发现, 但直到 20 世纪 90 年代, 四元数的理论和方法才被开始应用到彩色图像处理中<sup>[3]</sup>。到目前为止从事该部分内容研究的主要学术机构有美国西雅图大学、美国卡内基-梅隆大学以及国内的电子科技大学、厦门大学、浙江大学等。虽然科研人员在该领域已经开展的工作仍然处于初期研究阶段, 但提出的基于四元数的彩色图像边缘检测方法(如色彩差分、矢量点乘等)将一个彩色像素作为一个整体进行处理, 体现了它的先天优势, 而 1995 年提出的粒子群优化算法是一种基于迭代的优化工具。从它提出到现在, 已得到了学者的广泛关注, 并在许多方面进行了运用<sup>[4]</sup>。

将四元数理论和粒子群相结合引入彩色图像边缘检测中, 从已经公开发表的学术论文和相关的文献报道来看目前还没有人做过, 由于标准 PSO 算法本身也存在缺陷, 因此人们提出了各种改进粒子群优化算法, 这些算法各有优缺点。

本文提出了一种改进粒子群优化算法, 将其引入四元数表示的彩色图像中。

### 2 粒子群优化算法原理

粒子群优化(Particle Swarm Optimization, PSO)算法中的粒子指的是每个优化问题的解, 它被看作是  $d$  维搜索空间中一只鸟, 称为“粒子”, 如粒子  $x$  就是维度为  $d$  的空间粒子, 而粒子群是指  $n$  个粒子组成的群落(粒子允许重复), 简称粒子群。

首先系统初始化一组随机粒子群, 通过迭代搜寻最优值, 在每一次迭代过程中, 粒子通过跟踪粒子本身所找到的最优解(个体值  $p_{best_k}$ )与整个粒子群的最优解(全局极值  $p_{gbest_k}$ )更新自己的速度和位置, 从而产生新一代群体, 并计算由优化问题所决定的适应度函数值, 通过适应度值评价粒子的优劣, 不断迭代找到系统的最优解。其速度和位置更新公式如下:

$$v_{k+1} = wv_k + c_1 \times r_1 \times (p_{best_k} - x_k) + c_2 \times r_2 \times (p_{gbest_k} - x_k) \quad (1)$$

$$x_{k+1} = x_k + v_{k+1} \quad (2)$$

其中,  $v_k$  是粒子的速度矢量;  $x_k$  表示粒子的当前位置;  $p_{best_k}$  表示粒子本身所找到的最优解的位置;  $p_{gbest_k}$  表示整个种群当前所找到的全局最优解。文献[5]指出  $c_1=1.5$ ,  $c_2=2.5$ , 能使算法收敛性能更好,  $w$  是加权因子, 取值在 0.4~0.9 之间。

### 3 改进粒子群优化算法

由上述对粒子群算法的原理可知, 由于每个粒子在飞行过程中只注意到自身历史最优值和群体历史最优值, 无法判

**基金项目:** 成都理工大学研究基金资助项目(2008YQ01)

**作者简介:** 刘笃晋(1971—), 男, 硕士, 主研方向: 图像处理, 模式识别; 孙淑霞、丁照宇, 教授; 李思明, 讲师

**收稿日期:** 2011-01-21 **E-mail:** xszy138@126.com

别是否是全局最优值, 很容易陷入局部最优, 因此往往会遇到早熟收敛的问题。这些早熟收敛点, 有可能是局部极小点, 或是局部极小点邻域的一个点。也就是说, 早熟收敛并不能保证算法收敛到局部极小点, 并且, PSO 向最优点的收敛速度也比较缓慢。实际上对 PSO 的研究发现, PSO 早期收敛速度较快, 但到寻优后期, 寻优结果的改进则不尽理想。这主要归因于算法收敛到局部极小点时, 缺少有效机制使算法逃离局部极小点。针对这些问题, 出现了许多改进算法<sup>[6]</sup>, 考虑到本实验中的求解函数是高维函数, 且多个局部极值, 除非当粒子群恰巧初始化在全局最优值附近, 否则标准粒子群算法就很容易陷入局部最优的陷阱, 针对这个问题, 本文提出一种改进粒子群优化算法, 首先用下式:

$$\sigma^2 = \sum_{i=1}^n \left| \frac{f_i - f_{\text{avg}}}{f} \right| \quad (3)$$

判断是否存在早熟现象。其中,  $f_i$  是  $i$  个粒子的适应度;  $f_{\text{avg}}$  是平均适应度;  $f$  是归一化因子;  $\sigma^2$  是适应度的方差;  $\sigma_d^2$  的取值与实际问题的有关, 一般设为常数, 当  $\sigma^2 < \sigma_d^2$  时, 标志着出现早熟现象。为促使粒子逃离局部最优的陷阱, 最大限度地提升算法全局搜索的性能, 根据粒子群优化算法原理结合其他算子(高斯、柯西分布等)优点和本实验特点, 提出一种新的算子进行变异:

$$gbest_k = gbest_k + \frac{0.314\beta}{\sqrt{2\pi}(gbest_k^{3/2} + \beta^{3/2})} \quad (4)$$

其中,  $\beta=2$ 。这种算子进行变异比其他算子如高斯或柯西分布更容易产生一个远离原点的随机数, 且产生的随机数分布范围更大, 有利于粒子逃出局部最优的陷阱, 也容易收敛。

#### 4 基于四元数的图像边缘原理

四元数理论<sup>[7]</sup>从提出至今在图像处理中已得到了广泛应用。在四元数理论中, 若设  $q$  是四维空间中的一个数, 它包含 1 个实部  $a$  和 3 个虚部  $b$ 、 $c$  和  $d$ , 记这个四维空间的基为  $\{1, i, j, k\}$ , 则四元数  $q$  可以表示为:

$$q = a + bi + cj + dk$$

其中,  $a$ 、 $b$ 、 $c$ 、 $d$  是实数, 当实部  $a=0$  时, 称  $q$  为纯虚四元数(Pure quaternion)。记  $\text{Re}(q)=a$  表示  $q$  的实部,  $\text{Im}(q)=bi+cj+dk$  为  $q$  的虚部。于是, 有:

$$q = \text{Re}(q) + \text{Im}(q)$$

##### 4.1 四元数的极坐标

对于四元数  $q = a + bi + cj + dk$ , 令:

$$\mu = \frac{1}{\sqrt{b^2 + c^2 + d^2}}(bi + cj + dk)$$

$$\theta = \begin{cases} \tan^{-1}(\sqrt{b^2 + c^2 + d^2}/a) & a \neq 0 \\ \pi/2 & a = 0 \end{cases}$$

则经过计算有  $q = |q|(\cos\theta + \mu\sin\theta) = |q|e^{\mu\theta}$ 。其中,  $\mu$  是单位纯虚四元数(即模为 1 的纯虚四元数), 也称为四元数  $q$  的特征轴, 而  $\theta$  称为特征角。

设矢量  $\alpha$  旋转  $\theta$  角变成矢量  $\beta$ , 设垂直于  $\alpha$ 、 $\beta$  所在平面的转轴方向单位矢量为  $e_n$ , 转动方向与转轴方向遵守右手法则, 这时四元数的表示为:

$$\beta = (\cos\theta + e_n\sin\theta)\alpha$$

当反向旋转时, 相应公式为:  $\beta = (\cos(-\theta) + e_n\sin(-\theta))\alpha$ , 而当矢量  $\alpha$  与转轴  $e_n$  不垂直时, 这时若  $\alpha$  绕转轴转动  $\theta$  角变成  $\beta$  时, 此时四元数表示为:

$$\beta = (\cos\frac{\theta}{2} + e_n\sin\frac{\theta}{2})\alpha(\cos\frac{\theta}{2} - e_n\sin\frac{\theta}{2})$$

设 RGB 彩色图像的像素  $x_{ij}=[r_{ij}, g_{ij}, b_{ij}]^T$ , 其中,  $r_{ij}$ 、 $g_{ij}$ 、 $b_{ij} \in \{0, 1, \dots, 255\}$ 。为了对称, 将 RGB 空间的原点坐标从  $(0, 0, 0)$  移到 RGB 空间的中心  $(127.5, 127.5, 127.5)$ , 则表示彩色像素  $x_{ij}$  的纯虚四元数为:  $q_{ij}=(r_{ij}-127.5)i+(g_{ij}-127.5)j+(b_{ij}-127.5)k$  在三维空间内, 设  $\mu$  是一个用单位纯虚四元数表示的单位三维矢量, 用四元数极坐标形式表示为  $R = e^{\mu\theta} = \cos\theta + \mu\sin\theta$ , 根据上面四元数三维转动原理知  $RXR^*$  表示将三维矢量  $X$ (用纯虚四元数表示)绕轴  $\mu$  旋转  $2\theta$  角度。

#### 4.2 彩色图像边缘表示

在 RGB 三维空间中将彩色像素值当作矢量, 那么此矢量模值表示此像素的亮度, 其方向表示此像素的彩色色调和饱和度。若一个像素矢量的 3 个分量值相等, 则此像素矢量为灰度矢量。为了检测图像边缘, 以灰度像素矢量  $\mu=i+j+k$  作为旋转轴, 即当  $\theta = \pi/2$ , 以极坐标形式表示为  $R = e^{\mu\theta} = i+j+k$ , 那么对任一用纯虚四元数表示矢量  $X$ ,  $RXR^*$  表示将  $X$  绕单位矢量  $\mu$  旋转  $180^\circ$ , 这时实际上将  $X$  转到了以单位矢量  $\mu$  为对称轴的相反方向上, 再将  $X+RXR^*$  之后所得矢量三分量的值一定相等, 故必为灰度矢量。根据这个原理, 由于任一彩色像素矢量  $q$ , 都有  $RqR^*$  是一个纯虚四元数, 那么对任 2 个彩色像素矢量  $q_1=r_1i+g_1j+b_1k$  和  $q_2=r_2i+g_2j+b_2k$ , 令  $q_3=q_1+Rq_2R^*$ 。如果两彩色像素矢量相同, 则  $|q_1+Rq_2R^*|=0$ , 若  $q_1$  和  $q_2$  色调接近, 那么  $q_3$  必位于灰度矢量  $\mu$  附近, 即是说存在任意小的正数  $\delta$ , 有  $|q_1+Rq_2R^*|<\delta$ ( $\delta$  为任意小正数), 用此方法, 若此 2 个彩色像素矢量  $q_1$  和  $q_2$  在图像边缘, 自然也能判断出来。

#### 5 图像边缘的检测

按前面的方法, 对任何像素  $q=ri+gj+bk$ , 取一个灰度像素矢量  $\mu=i+j+k$  为旋转轴。为判断是否为边缘像素, 采用如下左右双模板做卷积操作:

$$\begin{bmatrix} \beta & \beta & \beta \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_1 & c_2 & c_3 \\ 0 & 0 & 0 \\ c_4 & c_5 & c_6 \end{bmatrix} \begin{bmatrix} \beta' & \beta' & \beta' \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (5)$$

其中,  $c_1$ 、 $c_2$ 、 $c_3$ 、 $c_4$ 、 $c_5$ 、 $c_6$  为图像中像素, 而  $\beta = \cos\frac{\theta}{2} + \mu\sin\frac{\theta}{2}$ ,  $\beta' = \cos\frac{\theta}{2} - \mu\sin\frac{\theta}{2}$ ,  $\mu=i+j+k$ , 此模板是将像素  $c_1$ 、 $c_2$ 、 $c_3$  绕灰度轴按右手法则旋转  $180^\circ$ , 再分别与  $c_4$ 、 $c_5$ 、 $c_6$  相加, 最后求总和, 结果如下式:

$$c'(m, n) = (\beta c_1 \beta' + c_4) + (\beta c_2 \beta' + c_5) + (\beta c_3 \beta' + c_6)$$

如此矢量三色分量相等或近似相等, 表示像素  $q$  为局部同色像素; 反之, 为边缘像素。

在粒子群优化算法中, 用一个像素表示为一个粒子中的一个元素  $\{q_{ij}\}_{i,j=1}^N$ , 一幅图像中用上述模板对测试图像进行卷积, 结果矢量三分量相等或近似相等设为 0; 反之, 设为 1。当全部边缘像素点都检测出来时, 适应度函数  $E_{\max}(q_{11}, q_{12}, \dots, q_{nn})$  达到最大, 适应度值为:

$$E_{\text{det}} = \begin{cases} 0 & \text{若 } c'(m, n) \text{ 三色分量相等或近似相等} \\ 1 & \text{其他} \end{cases} \quad (6)$$

$$E_{\text{max}} = \sum_{i=0}^n E_{\text{det}} \quad (7)$$

改进 PSO 算法进行图像边缘检测的基本步骤:

(1) 初始化粒子群, 设定粒子数为  $n_p$ , 随机设置粒子的初始位置  $v_0^k$  和速度  $x_0^k$  ( $1 \leq k \leq n_p$ ), 每个粒子  $p_{\text{best}_0}$  设为初始位置, 全局最好位置设为  $p_{\text{gbest}_0}$ , 最大迭代次数为  $n_{\text{max}}$ , 迭代计数器  $n_c=0$ 。

(2) 计算每个粒子的适应度值  $E^k(q_{11}, q_{12}, \dots, q_{nn})$ 。

(3) 对每个粒子, 比较它的适应度值和其经历的最好位置  $p_{best_k}$ ; 若更好, 更新  $p_{best_k}$ 。

(4) 对每个粒子, 比较它的适应度值和群体所经历的最好位置  $p_{gbest_k}$ ; 若更好, 更新  $p_{gbest_k}$ 。

(5) 根据位置和速度的更新公式(1)和式(2)调整粒子的位置和速度。

(6) 根据计算是否早熟, 若早熟用变异算子改变  $p_{gbest_k}$ , 然后转到步骤(2)。

(7) 若达到结束条件(适应度函数不再增加或最大迭代次数), 结束; 否则, 转步骤(2)。

(8) 输出全局最优解  $gbest$ , 算法结束。

## 6 实验结果与分析

实验运行环境为: CPU Intel i5 750, Matlab7.0。采用从 Internet 上下载的图片进行测试。粒子群算法的参数设置为: 粒子数设定为  $n_p=60$ , 惯性因子  $\omega_{\max}=0.9$ ,  $\omega_{\min}=0.4$ , 加速常数  $c_1=1.4$ ,  $c_2=1.6$ , 对利用 sobel 算子的标准彩色图像边缘检测算法(简称方法 1)、基于四元数的标准粒子群彩色图像边缘检测算法(简称方法 2)和基于四元数的改进粒子群彩色图像边缘检测算法(简称方法 3)进行了 10 次实验, 结果如表 1 所示, 其对应图像如图 1~图 4。图 1 是原始图像图, 图 2~图 4 分别是方法 1、方法 2、方法 3 的结果图。

表 1 3 种方法的边缘检测时间

实验次数	方法 1	方法 2	方法 3
1	14.02	7.92	6.48
2	14.22	8.73	6.23
3	13.89	8.31	5.98
4	13.78	8.62	5.56
5	13.19	8.22	5.41
6	13.56	8.11	5.86
7	13.68	8.31	6.21
8	13.27	8.42	6.27
9	13.78	8.89	5.87
10	13.02	8.12	6.02
均值	13.641	8.365	5.989



图 1 原始图像



图 2 方法 1 检测结果



图 3 方法 2 检测结果



图 4 方法 3 检测结果

从检测的图像结果可以看出, 方法 1 中由于是采用的传统彩色边缘检测方法, 其像素三色并不是作为一个整体进行处理的, 割裂了彩色图像的 RGB 的 3 个通道之间固有的强烈的光谱联系, 并且检测时只考虑了一个像素和其邻近像素关系, 没考虑到像素的全局作用, 因而检测效果很不理想, 存在边缘丢失(如图中女孩的脸边沿), 一些重要的边缘细节信息无法检测出(如头发纹理), 还存在明显的噪声干扰。

方法 2 和方法 3 运用了四元数来处理像素和 PSO 算法的引入, 具有更强的敏感性, 能够提取出的边缘细节要远多于方法 1, 很好地提取出了该类彩色图像的边缘信息, 如图中女孩的鼻梁、嘴巴、甚至头发纹理等, 并且在边缘上的邻近像素点出现了不连续现象。

方法 1 没有引入粒子群, 无法通过整体最优来判断是不是边缘, 因而会出现错判, 甚至出现边缘丢失。而方法 2 和方法 3 本身由于它们都是将一个图像像素作为一个整体进行处理的, 都是将粒子群优化算法引入其中的, 因而它们的图像边缘检测结果相比没有非常明显的差异。

从检测的速度来看, 在平均时间方面方法 2 比方法 1 快了 5 s 多, 而方法 3 又比方法 2 快了 2 s 多, 这主要由于改进 PSO 算法采用的一种新的变异算子对每个粒子的速度和群体最优位置进行变异, 促使陷入局部最优的粒子分散开来, 然后重新在新的解空间中飞行, 直到找到最优解或者达到最大迭代次数而终止, 因而更容易收敛, 收敛时间更短。本文提出的改进算法克服了标准 PSO 算法的早熟现象, 这充分说明了改进算法在收敛精度、速度等方面都能达到一个较合理的状态。

## 7 结束语

本文采用四元数矢量旋转原理, 利用四元数表示图像矩阵, 将一种改进粒子群优化算法引入, 提出了一种新的基于四元数粒子群优化算法的图像边缘检测方法。在检测图像边缘的细微色彩变化、图像纹理细节以及图像边缘的速度、精度方面, 本文提出的方法都比传统彩色图像边缘检测方法对图像边缘检测效果好。实验证明, 该方法能够满足实际需要。

## 参考文献

- [1] How Z J, Wei G W. A New Approach to Edge Detection[J]. Pattern Recognition, 2002, 35(7): 1559-1570.
- [2] 宋莹, 陈科, 林江莉, 等. 基于图像分块的边缘检测算法[J]. 计算机工程, 2010, 36(14): 196-197.
- [3] Cai C, Miran S K. A Normalized Color Difference Edge Detector Based on Quaternion Representation[C]//Proc. of IEEE International Conference on Image Processing. Vancouver, Canada: [s. n.], 2000: 816-819.
- [4] 刘希玉, 刘弘. 神经网络与微粒群优化[M]. 北京: 北京邮电大学出版社, 2008.
- [5] Pudil P, Novovicova J, Kittler J. Floating Search Methods in Feature Selection[J]. Pattern Recognition Letters, 1994, 15(11): 1119-1125.
- [6] 于雪晶, 麻肖妃, 夏斌. 动态粒子群优化算法[J]. 计算机工程, 2010, 36(4): 193-194.
- [7] 李文亮. 四元数矩阵[M]. 长沙: 国防科技大学出版社, 2002.

编辑 顾逸斐