

一种基于四元数的彩色图像边缘检测改进算法

黄伟, 周鸣争, 李小牛

(安徽工程科技学院 计算机科学与工程系, 安徽 芜湖 241000)

摘要: 图像边缘检测对后续的图像分割和识别具有重要的作用。针对彩色图像的边缘检测的实际需求, 对比分析了经典边缘检测算子的特点和不足, 提出了多方向的 Sobel 边缘检测算子模板, 并且针对传统边缘检测算法处理速度慢、运算量较大、对边缘细节位置处理效果差等缺点, 结合彩色图像的四元数描述方法提出改进算法, 结合对颜色空间的分解实现了对彩色图像的边缘检测。实验证明算法是有效的, 边缘检测效果好且易于实现, 使用四元数描述方法有效提高了边缘检测的速度。

关键词: 彩色图像; Sobel 算子; 边缘检测; 四元数描述

中图分类号: TP391.41

文献标识码: A

文章编号: 1673-629X(2008)03-0121-04

A Color Image Edge Detection Modified Algorithm Based on Quaternion Representation

HUANG Wei, ZHOU Ming-zheng, LI Xiao-niu

(Department of Computer Science and Engineering, Anhui Univ. of Tech. and Science, Wuhu 241000, China)

Abstract: Image edge detection is very important to the subsequent image segmentation and recognition. Aiming at the actual application of color image edge detection, some characteristics and faults of the traditional edge detecting template are compared and analyzed. A multidirection Sobel edge detection template is proposed. Aiming at the faults of the traditional algorithms, the modified algorithm is proposed with the help of quaternion representation. The essay carries out color image edge detection by decomposing color space. The result shows that the algorithm is effective and simple. The speed of edge detection is improved with the method of quaternion representation.

Key words: color image; Sobel template; edge detection; quaternion representation

0 引言

边缘是数字图像的重要特征。边缘检测是图像处理过程中的重要内容, 它是后续图像分割、特征提取等的基础, 边缘检测的原理是检测每个像素点及其领域的状态, 以确定像素点是否处于边界上。常见的边缘检测算子有 Roberts 算子、Sobel 算子、Prewitt 算子和 Canny 算子^[1]等, 但上述算子大多主要应用于灰度图像。有研究表明^[2], 彩色图像中大约有 90% 边缘与灰度图像中的边缘相同, 因此可将以上算子直接应用到彩色图像上, 但是通常边缘检测效果不佳, 一些重要的细节边缘信息无法检测出, 而且这种方法容易受噪声干扰。文中介绍的改进算法在分解后的 RGB 空间中, 应用 8 方向上的 Sobel 边缘检测模板, 取得了较好的边

缘检测效果, 改进后的算法增强了边缘且抗噪性强。

1 基于 Sobel 算子的彩色图像边缘检测改进算法

1.1 边缘检测梯度算子

数字图像的一阶导数是基于各种二维梯度的近似值。图像 $f(x, y)$ 在点 (x, y) 梯度是如下的向量: $\text{Grad} = \begin{bmatrix} G_x \\ G_y \end{bmatrix}$, 其中 G_x, G_y 分别是图像函数 $f(x, y)$ 在每个像素点 (x, y) 处的两个偏导数。梯度方向是指点在 (x, y) 处的 f 最大变化率方向。在边缘检测整个实现过程中, 寻找的就是图像中梯度方向上的像素点。

常见的边缘检测梯度算子有 Roberts 算子、Sobel 算子、Prewitt 算子等。图 1 中显示的 3×3 大小窗口区域表示图像中某像素的领域, Roberts 算子实现比较简单, 对具有陡峭的低噪声图像处理效果比较好, 但对边缘定位不是很准确。得到中心点处的一阶偏导数的方法可使用图 1 中所示的 Roberts 梯度算子:

收稿日期: 2007-06-15

基金项目: 安徽省教育自然科学基金重点资助项目(2004kj033dz)

作者简介: 黄伟(1981-), 男, 安徽芜湖人, 硕士研究生, 助教, 研究方向为数字图像处理与模式识别; 周鸣争, 教授, 硕士生导师, 研究方向为图像处理与模式识别、计算机网络。

$$G_x = (P_9 - P_5) \quad G_y = (P_8 - P_6)$$

P_1	P_2	P_3
P_4	P_5	P_6
P_7	P_8	P_9

图 1 Roberts 算子

用图 1 中所示的算子模板结合近似差分可以得到图像 $f(x, y)$ 的导数, 表示如下:

$$G_x = (P_7 + P_8 + P_9) - (P_1 + P_2 + P_3)$$

$$G_y = (P_3 + P_6 + P_9) - (P_1 + P_4 + P_7)$$

即在 3×3 模板区域中, 将第 1 行与第 3 行之差近似为 x 方向上的导数, 第 3 列与第 1 列之差近似为 y 方向上的导数。图 2 所示为 Prewitt 算子。它包含 x 和 y 两个方向的模板算子, 可以分别使用水平算子和垂直算子对图像进行卷积, 更改 Prewitt 算子模板中心系数上的权值, 即得到图 3 所示的 Sobel 算子, Sobel 算子也包含水平模板和垂直模板:

$$G_x = (P_7 + 2P_8 + P_9) - (P_1 + 2P_2 + P_3)$$

$$G_y = (P_3 + 2P_6 + P_9) - (P_1 + 2P_4 + P_7)$$

-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1

图 2 Prewitt 算子

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

图 3 Sobel 算子

Prewitt 算子和 Sobel 算子对渐变图像和噪声较多的图像处理效果较好, 但是相比较而言, Sobel 算子对边缘定位较为准确, 且抗噪能力较强。所以 Sobel 算子在实际应用中常被采用, 将其应用于彩色图像边缘检测时, 算法实现较简单但通常边缘检测效果欠佳, 存在不能很好地检测出细节边缘, 甚至漏检的缺点。

1.2 不同颜色空间下的边缘检测对比研究

在彩色图像的边缘检测过程中, 颜色空间的选择首先是一个必须要考虑的问题。目前存在多种颜色空间类型^[3], 如 RGB, HSI, CMY, YIQ, YUV, CIE YUV, HSL 等。同时对应于不同的颜色空间也出现了很多的边缘提取方法:

(1) 一般彩色图像都是以 RGB 方式存储的, RGB 颜色空间适合于显示系统, 但它不能较好地反映和描述人类的视觉感知。此空间上的边缘检测通常是先将彩色图像的像素分解为 R、G、B 三个基本颜色分量, 然后通过对 3 个分量进行边缘检测的模板运算并进行综合, 最终完成检测过程。因为没有对颜色空间进行线

性或非线性转换, 所以免去了较大的计算量, 检测结果也不会歪曲原有的颜色信息。大量实验和研究证明^[3], 在 RGB 空间中进行边缘检测能取得次最优并接近最优的结果。

(2) 基于 HSI 空间的彩色边缘检测。HSI 即色度 (Hue)、亮度 (Intensity)、饱和度 (Saturation), 其中 H 和 S 分量为颜色信息通道, 而 I 分量与颜色信息无关。经过变换, H、S、I 三分量之间的相关性减小, 因此亮度和色度的分离也更有利于图像处理^[4], HSI 空间中亮度和色度分离可用于边缘检测, 特别是在一些明暗对比不均的场合下, 此空间中边缘检测效果较好, 能提取更多的彩色边缘信息并具有很好的抗噪声性能。但是最大的不足在于空间转换的计算量很大, 不能达到实时处理的要求。

以上两种主要的颜色空间使用比较多。经过比较研究, 文中所提出的算法选择在 RGB 颜色空间中对彩色图像进行多分量的边缘检测处理。

1.3 彩色图像的四元数描述法

灰度图像边缘检测方法应用到彩色图像上时, 往往并不能产生比较好的边缘提取效果, 而且会在算法实现过程中产生诸多问题, 以至影响到算法执行的效率和边缘提取的效果。究其原因, 主要是没有找到合适的方法来正确全面地表达彩色图像。彩色图像含有大量丰富的图像信息, 当然包括我们所感兴趣的边缘信息, 如何能高效地提取和检测出边缘信息, 首先关键要解决好彩色图像信息的描述和表达的问题, 大多数方法将一幅彩色图像看成是矢量的集合, 在此基础上应用各种算法实现了彩色图像的边缘提取, 但是基于矢量的算法都存在算法实现复杂、运算量大的缺点, 这在很大程度上制约了最终边缘提取和提取的效果。

近几年来, 基于四元数描述的彩色图像处理方法大量被应用^[5], 并且取得了比传统方法更好的处理效果。哈密尔顿在复数的基础上对实数进行了进一步的推广, 定义了四元数的形式, 它由一个实部和三个虚部构成, 三个虚部彼此正交。四元数 $P = w + xi + yj + zk$, i, j, k 均是虚数单位。也就是说, 四元数描述形式其实是标量加上矢量。另外规定了如下规则:

$$i^2 = j^2 = k^2 = ijk = -1$$

$$ij = k, jk = i, ki = j,$$

$$ji = -k, kj = -i, ik = -j$$

因为一个四元数有 4 个部分, 所以可以简单表示为 $P = (a_0, a_1, a_2, a_3)$, 其中 a_1, a_2, a_3 为四元数对应的三个虚部, 满足矢量运算法则。四元数的乘法运算很重要, 定义两个四元数 $P1 = (a_0, a_1, a_2, a_3)$ 和 $P2 = (b_0, b_1, b_2, b_3)$ 的乘积为:

$$P_1 P_2 = (a_0 + a_1 i + a_2 j + a_3 k) \times (b_0 + b_1 i + b_2 j + b_3 k) = (a_0 b_0 - a_1 b_1 - a_2 b_2 - a_3 b_3, a_0 b_1 + a_1 b_0 + a_2 b_3 - a_3 b_2, a_0 b_2 - a_1 b_3 + a_2 b_0 + a_3 b_1, a_0 b_3 + a_1 b_2 - a_2 b_1 + a_3 b_0) \quad (1)$$

在实际表达和应用时,可以把四元数形式简化为 $Q = (a_0, I)$, 其中 $I = a_1 i + a_2 j + a_3 k$ 。简化形式也就是将四元数描述形式划分为标量部分和矢量部分,此时两个四元数 P_1 和 P_2 的乘积可写为:

$$P_1 P_2 = (a_0 b_0 - I_1 \cdot I_2, a_0 I_2 + b_0 I_1 + I_1 \times I_2) \quad (2)$$

如果四元数中实部 a_0 为零,则称其为纯四元数^[5],通常记为 L 。两个纯四元数 L_1 和 L_2 相乘可表示为: $L_1 L_2 = (-L_1 \cdot L_2, L_1 \times L_2)$ 。一个四元数 P 的模可定义为下式:

$$|P| = \sqrt{a^2 + b^2 + c^2 + d^2} \quad (3)$$

对四元数 P 可以通过归一化为 $|P_0| = |P|/|P| = 1$,从而转换为单位四元数,类似于向量的标准化。当 a_0 等于零时,一幅 RGB 空间下的彩色图像可以用以下的四元数表示为:

$$p = ri + gj + bk \quad (4)$$

其中, r, g, b 分别表示红、绿、蓝 3 种颜色分量的亮度值,这种表示形式不同于以往的彩色图像描述方式,它能简洁并更好地表达彩色图像信息。

1.4 基于多方向 Sobel 算子模板的彩色图像边缘检测改进算法

文中算法针对于彩色图像,将其在 RGB 颜色空间下分解成 R, G, B 三幅分量图像。经典的 Sobel 边缘检测算子只有水平和垂直两个方向的模板,其模板的方向仅表示灰度由低到高(或由高到低)的依次变化方向,而不是图像的实际边缘方向,处理后容易使得大量有用的颜色信息丢失。因此文中以 Sobel 算子为基础提出改进算法,定义了 8 方向上的模板算子(3×3 窗口),并使每个算子模板的方向更能体现图像的实际边缘方向,这样可以更好地提取出包含大量颜色信息的边缘^[6]。改进的 Sobel 边缘检测模板算子如图 4 所示。

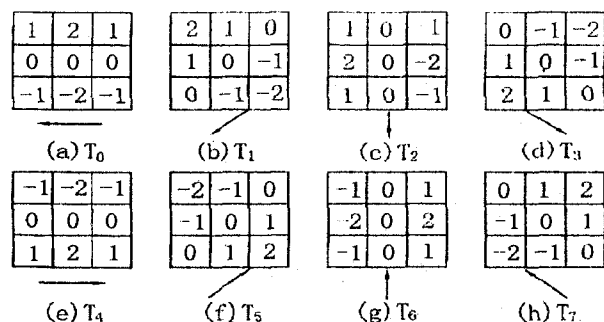


图4 改进的 Sobel 边缘检测模板算子

如图所示, T_0 到 T_7 为扩充改进后的 8 个模板方向,分别用这 8 个模板对彩色图像进行扫描计算,并取结果中的最大值作为该像素点的梯度值,并定义最大值对应模板的方向顺时针转 90° 为梯度方向。这样就完成了边缘检测的模板运算。具体算法如下:

(a) 读入原始图像 $G(x, y)$ 。

(b) 将 $G(x, y)$ 在 RGB 空间上分解成 $RGBX(x, y)$, $RGBY(x, y)$, $RGBZ(x, y)$ 三幅图像。

(c) 对于三幅图像,分别用改进的 Sobel 模板算子对图 1 中的像素点 P_5 做以下运算:

$$E_0 = P_1 + 2P_2 + P_3 - P_7 - P_8 - P_9$$

$$E_1 = 2P_1 + P_2 + P_4 - P_6 - 2P_9 - P_8$$

.....

$$E_7 = P_2 + 2P_3 + P_6 - P_8 - 2P_7 - P_4$$

E_0 到 E_7 为 8 方向模板的运算结果,取 $E(i)$ ($i = 0, 1, 2, \dots, 7$) 中的最大值 $E(\max)$ 为像点最后的梯度值。

(d) 综合三幅子图像上的检测结果,得到最终的边缘检测图像 $E(x, y)$ 。

上述改进算法在实现过程中,利用 8 方向上的模板算子,提取出彩色图像中比较丰富的边缘信息,算法实现效果较好。

2 实验结果讨论与分析

文中算法中选取一尺寸为 522×390 的彩色图像 house.bmp,在 Delphi 7.0 平台下对其使用文中的改进算法,实验结果如图 5~图 10 所示。



图5 原图



图6 直接应用 Sobel 算子检测结果



图 7 R 分量图



图 8 G 分量图



图 9 B 分量图

图 5 为原彩色图像;图 6 是直接利用 Sobel 算子得到的结果,由结果图可以看出有些主要边缘信息没有

被检测出,而且图中房屋、树木等多处边缘不连续;图 7、图 8 和图 9 为 RGB 空间分量图像。分析实验结果,与传统算法相比较,文中提出的彩色图像边缘检测改进算法能较好并更多地保留原有图像的颜色和轮廓特征,算法实现简单,实现效果较好,但是原有图像中的一些细节信息缺失,这有待进一步对算法进行改进。

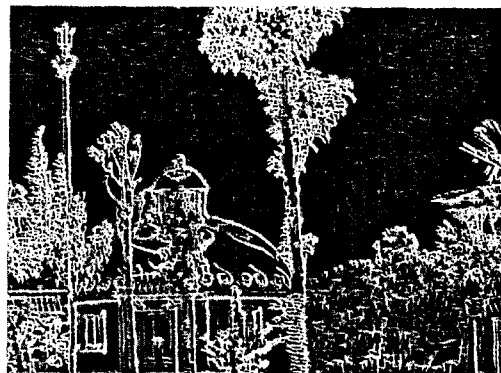


图 10 文中算法检测结果图像

参考文献:

- [1] Castleman K R. Digital Image Processing[M]. 北京:清华大学出版社, 2003.
- [2] Health A, Sarkar S, Sanocki T, et al. Comparison of Edge Detectors: A Methodology and Initial Study[J]. Computer Vision and Image Understanding, 1998, 69(1): 38-54.
- [3] Dony R D, Wesolkowski S. Edge Detection on Color Images using RGB Vector angle[C]//IEEE Canadian Conference on Electrical and Computer Engineering. [s. l.]: [s. n.], 1999: 687-692.
- [4] Chen H C, Chien W J, Wang S J. Contrast-based Color Image Segmentation[J]. IEEE Signal Processing Letters, 2004, 11(7): 641-644.
- [5] Cai C, Miran S K. A Normalized color Difference edge Detector Based on Quaternion Representation[C]//In: Proceedings of IEEE International Conference on Image Processing. Vancouver, Canada: [s. n.], 2000: 816-819.
- [6] 章毓晋. 图像分割[M]. 北京: 科学出版社, 2001.

(上接第 120 页)

工作特点及相关细节就显得尤为重要,有了对 NAND FLASH 的深入了解,碰到问题时就可以从容面对了。

参考文献:

- [1] 马丰玺, 杨 斌, 卫洪春. 非易失存储器 NAND Flash 及其在嵌入式系统中的应用[J]. 计算机技术与发展, 2007, 17(1): 203-205.
- [2] 周 涛, 刘有源. 基于 MIPS 的 Au 1500NC 板上的启动程序 U-Boot 设计[J]. 计算机技术与发展, 2006, 16(8): 94

- 96.

- [3] Hynix, Intel Corporation, Micron Technology Inc. Open NAND Flash Interface Specification[R]. America: Hynix, 2006.
- [4] 刘洲洲, 张 捷. 基于嵌入式系统的 NANDFlash 坏块处理和 FMM 实现[J]. 计测技术, 2006(6): 42-44.
- [5] Hynix. 4Gb NAND FLASH HY27UT084G2M[R]. Korea: Hynix, 2006.
- [6] Hynix. NAND Flash PART NUMBERING[R]. Korea: Hynix, 2006.