

一种去除图像混合噪声的快速 PGF 滤波

A Quick Peer Group Filtering for Mixed Noise Removal in Image

曹 刚 游志胜 刘直芳 徐 欣

(四川大学计算机系图形图像研究所 成都610064)

Abstract We present in the paper a quick and efficient technique for mixed noise removal in images based on the peer group filtering (PGF). This method first compute local energy of each image pixel. Center pixels whose "energy" exceeds the local threshold are replaced with the local marginal median. Other pixels come into operation of PGF. When classifying the peer group of pixel, we directly take the neighbor pixels which lies in direction of minimum feature changing as the peer group member instead of Fisher's linear discriminant. The experiment results show that proposed method which greatly improved speed of PGF, efficiently not only remove mixed Gaussian and impulse noise in image, but also preserve image edge information.

Keywords Peer Group Filtering, Fisher discriminant, Local energy

1. 引言

在图像处理的预处理阶段中,噪声的滤除往往非常重要。高斯滤波和中值滤波是最常用的滤波方法。其中高斯滤波对滤除高斯噪声非常有效,但会损坏图像的边缘信息,从而使图像变得模糊起来。而中值滤波能很好地滤除脉冲噪声,但对消除高斯噪声则效果不佳。两者共同的缺陷是对滤波窗口中所有邻域像素都采用一致的处理,从而会引入误差,损坏图像的边缘和细节。于是能快速地滤除混合噪声,又能保护好图像边缘信息的自适应滤波方法成为图像滤波中主要的研究热点。

近年来,针对上述问题提出了许多有效的滤波方法^[2~7],其中较为显著的是 Y. Deng 和 C. Kenney^[4~6]等提出的一种称为 PGF (Peer Group Filtering) 的滤波方法。该方法主要思想是只找出滤波窗口中与中心像素距离特征相近的邻域像素作为同组成员参加滤波。它是先根据滤波窗口中邻域像素与原像素特征距离值进行升序排列,再通过 Fisher 判别找出该像素的同组成员,即与中心像素特征值最相近的点。然后用滤波窗口中属于同组成员像素的加权特征值代替原来中心像素特征值。PGF 滤波克服了以往高斯滤波和中值滤波的滤波窗口中所有像素都参与滤波的弊端,从而能很好地滤除了混合噪声,又保护了图像的边缘信息。但是 PGF 滤波在归类每个像素的同组成员时需要计算距离、排序、Fisher 判别等大量的数学运算,特别是滤波窗口较大时,其滤波的速度极慢,因而很不适合于实时图像处理。本文提出的改进方法是先通过计算图像像素局部能量的方法去除脉冲噪声^[7],再在对邻域像素进行同组成员归类时,不使用 Fisher 判别,而是直接将像素特征梯度的垂直方向,即特征变化最小方向^[8]的邻域像素归为该像素的同组成员,最后再用同组成员进行滤波。该方法大大提高了以往 PGF 滤波的运算速度,有效地滤除了含高斯和脉冲的混合噪声,又很好地保护了图像边缘信息。

2. PGF 滤波

PGF 滤波主要分确定原像素同组成员和滤波两个阶段。要确定原像素的同组成员,首先要计算在滤波窗口 $w \times w$ 中各邻域像素与原像素的特征向量的距离,然后进行升序排列。设 $P_0(x, y)$ 表示原像素 (x, y) 的特征向量,则滤波窗口中各像素与之的特征距离为:

$$d_i(x, y) = \|P_0(x, y) - p_i(x, y)\| \quad i=0, \dots, w \times w - 1 \quad (1)$$

将 $d_i(x, y)$ 进行升序排列:

$$d_0(x, y) \leq d_1(x, y) \leq \dots \leq d_{w \times w - 1}(x, y) \\ i=0, \dots, w \times w - 1 \quad (2)$$

然后由排序后的距离值根据 Fisher 判别来确定原像素 (x, y) 同组大小 s (即同组成员的个数)。 s 对应应 Fisher 准则函数取得极大值的 i , Fisher 准则函数定义如下:

$$J(i) = \frac{|m_1(i) - m_2(i)|^2}{s_1^2(i) + s_2^2(i)} \quad i=1, \dots, w \times w - 1 \quad (3)$$

其中距离均值:

$$m_1(i) = \frac{1}{i} \sum_{j=0}^{i-1} d_j(x, y) \quad m_2(i) = \frac{1}{w \times w - i} \sum_{j=i}^{w \times w - 1} d_j(x, y) \quad (4)$$

类内离散度:

$$s_1^2(i) = \sum_{j=0}^{i-1} |d_j(x, y) - m_1(i)|^2 \\ s_2^2(i) = \sum_{j=i}^{w \times w - 1} |d_j(x, y) - m_2(i)|^2 \quad (5)$$

通过(3)式计算各个 i 的 Fisher 准则函数值,便可由下式确定像素 (x, y) 的同组大小 s :

$$J(s) = \max J(i) \quad i=1, \dots, w \times w - 1 \quad (6)$$

为了消除滤波窗口中噪声的影响,可通过计算 $d_i(x, y)$ 的一阶差分 $f_i(x, y)$ 来判断脉冲噪声,即若:

$$f_i(x, y) = d_{i+1}(x, y) - d_i(x, y) > \alpha \quad (7)$$

则认为对应的像素点为脉冲噪声(α 为调节参数),并由剩下的其它像素来确定同组成员,具体做法见文[4]。确定了像素

曹 刚 博士研究生,主要研究方向为计算机视觉、图像处理与模式识别。游志胜 教授,博士生导师,主要研究领域:MTMST 系统、数据融合、图像处理与模式识别、神经网络、ITS 等。刘直芳 博士研究生,主要从事图像处理、模式识别和智能交通方面的研究。徐 欣 工程师,主要从事图像处理与模式识别研发工作。

(x, y) 的同组的大小 s , 即确定了同组的成员像素。于是便可用同组成员像素的加权特征向量值替代原像素的特征向量值, 从而完成 PGF 滤波。

从上可以看出, PGF 滤波在进行归类同组成员时, 包括计算特征距离、排序、Fisher 判别等大量的数学运算。其中 Fisher 判别过程是一个三重循环, 其计算复杂性为 $O(L^3)$ (L 为滤波窗口的大小), 需要耗费计算机大量运算时间。另外 PGF 滤波在用特征距离的一阶差分判断脉冲噪声时, 参数 α 很难调节, 容易造成误差。针对 PGF 滤波以上的缺陷, 本文提出下述的改进方法。

3. 改进的快速 PGF 滤波

3.1 根据局部能量滤除脉冲噪声

基于像素局部能量判断脉冲噪声是一种很有效的滤除脉冲噪声的方法, 像素的局部能量定义如下:

$$E(x, y) = \max(E_1(x, y), E_2(x, y)) \quad (8)$$

$$\begin{aligned} \text{其中: } E_1(x, y) = & |2 * (p(x, y) - \mu)^2 \\ & - (p(x-1, y) - \mu) * (p(x+1, y) - \mu) \\ & - (p(x, y-1) - \mu) * (p(x, y+1) - \mu)| \end{aligned} \quad (9)$$

$$\begin{aligned} E_2(x, y) = & |2 * (p(x, y) - \mu)^2 \\ & - (p(x-1, y-1) - \mu) * (p(x+1, y+1) - \mu) \\ & - (p(x+1, y-1) - \mu) * (p(x-1, y+1) - \mu)| \end{aligned} \quad (10)$$

式中 μ 表示滤波窗口中像素的局部均值。然后再计算局部阈值:

$$Th = \frac{a}{N} \times \sum_{n=1}^N E_n \quad (11)$$

式中 E_n 表示 N 个邻域像素中第 n 个像素的局部能量值, a 是经验常数。于是像素 (x, y) 是否为脉冲噪声由下面条件判断:

$$E(x, y) > Th \quad (12)$$

即 (x, y) 满足 (12) 则为脉冲噪声, 可作中值滤波处理。否则不为脉冲噪声, 作快速的 PGF 滤波处理。

3.2 同组成员的归类

为了提高运算速度, 在像素邻域进行同组成员归类时, 不再使用 Fisher 判别, 而是直接将与像素特征梯度垂直方向, 即特征变化最小方向的邻域像素归为该像素的同组成员。通常像素的特征梯度方向 η 可由下式计算:

$$\eta = \arctan(\Delta_x p(x, y) / \Delta_y p(x, y)) \quad (13)$$

$$\text{式中: } \Delta_x p(x, y) = p(x+1, y) - p(x, y) \quad (14)$$

$$\Delta_y p(x, y) = p(x, y+1) - p(x, y) \quad (15)$$

η 即表示像素特征变化最大方向, 而相应特征变化最小方向 ξ 则为:

$$\xi = \eta + \pi/2 \quad (16)$$

文[9]指出使用上述方法计算像素特征向量的梯度方向并不准确, 可使用下面图像差分乘方矩阵提取像素特征向量的梯度方向:

$$(dI)^2 = \left(\frac{dx}{dy} \right)^T \begin{bmatrix} \left(\frac{\partial I}{\partial x} \right)^2 & \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} & \left(\frac{\partial I}{\partial y} \right)^2 \end{bmatrix} \left(\frac{dx}{dy} \right) \quad (17)$$

$$\text{式中: } \frac{\partial I}{\partial x} = \Delta_x p(x, y) \quad \frac{\partial I}{\partial y} = \Delta_y p(x, y) \quad (18)$$

$$\text{令: } g_{11} = \frac{\partial I}{\partial x} \cdot \frac{\partial I}{\partial x} \quad g_{12} = \frac{\partial I}{\partial x} \cdot \frac{\partial I}{\partial y} \quad g_{22} = \frac{\partial I}{\partial y} \cdot \frac{\partial I}{\partial y} \quad (19)$$

则矩阵 $[g_{ij}]$ 的两个特征值就是 $\|dI\|^2$ 的两个极值, 对应的两个特征向量 η, ξ 就是图像在这点的两个特征变化方向:

$$\begin{cases} \lambda_{\pm} = \frac{g_{11} + g_{22} \pm \sqrt{(g_{11} - g_{22})^2 + 4g_{12}^2}}{2} \\ \eta = \frac{1}{2} \arctan \frac{2g_{12}}{g_{11} - g_{22}} \\ \xi = \eta + \frac{\pi}{2} \end{cases} \quad (20)$$

λ_+ 和 λ_- 这二个特征值表示了图像在这一点上的特征变化量的大小, 它们分别表示最大和最小的一个变化量, 而对应的变化方向则由 η 和 ξ 表示。

通过计算式 (20) (无需计算 $\lambda_{\pm}$), 可以求得像素 (x, y) 的特征变化最小的方向 ξ , 于是像素 (x, y) 的同组成员便由滤波窗口中 ξ 方向上的邻域像素组成。

3.3 据 ξ 决定的同组成员进行滤波

像素 (x, y) 同组成员归类后, ξ 方向上的邻域像素便组成该像素的同组成员, 即像素 (x, y) 的同组大小 s 也就确定了。于是便可用同组成员像素的加权特征向量值替代原像素的特征向量值, 即

$$P_{new}(x, y) = \sum_{i=0}^{s-1} w_i p_i(x, y) / \sum_{i=0}^{s-1} w_i \quad (21)$$

其中 $P_i(x, y)$ 属于 (x, y) 的同组成员像素的特征向量, w_i 是相应高斯滤波的加权系数。由 (21) 式可以看出, PGF 滤波实际是一种二进制的高斯滤波, 其中 1 表示该邻域像素是原像素的同组成员将参与滤波, 0 表示不是同组成员而不参与滤波。

从以上可以看出, 改进的算法主要是计算像素特征的梯度垂直方向 ξ 替代计算 Fisher 判别, 从而使计算复杂性从 $O(L^3)$ 减至 $O(L)$ 。综上, 改进的 PGF 滤波算法可由图 1 表示。

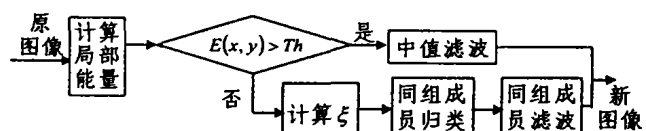


图1 改进的 PGF 滤波算法

4. 实验结果

本文就上述提出的改进的快速算法给出实验结果 (图 2), 并与滤波窗口大小相同的高斯滤波、中值滤波和原 PGF 滤波相比较, 以证明本算法的快速和有效性。其中图 (a) 为 256×256 的原图, (b) 为加了高斯噪声和脉冲噪声的图像, (c) 为高斯滤波的结果, (d) 为中值滤波的结果, (e) 为原 PGF 滤波的结果, (f) 为改进的快速 PGF 滤波的结果。比较 (b)、(c)、(d)、(e) 和 (f) 可以看出, 高斯滤波能较好地消除高斯噪声, 并同时平滑了图像的边缘, 使图像变得模糊起来。中值滤波较好地消除脉冲噪声, 却对高斯噪声无法很好地消除。原 PGF 滤波可较好地同时滤除高斯噪声和脉冲噪声, 并很好地保护了图像的边缘信息, 但是运算速度极慢。改进的快速 PGF 滤波也能达到 (e) 同样的效果, 但运算速度却大大地加快。以上实例都是采用 VC 在 PII 微机上完成, 其中原 PGF 算法需 1 分钟以上, 而改进的快速 PGF 却只需 2 秒以下。由此证明了改进的 PGF 滤波的有效性和快速性。

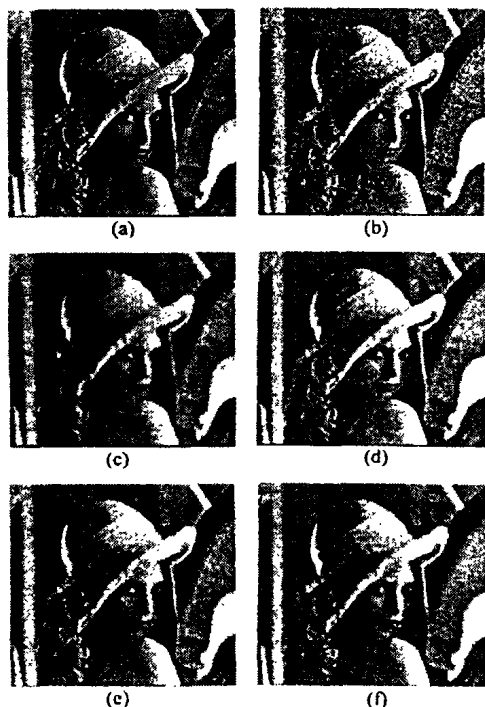


图2 实验结果

结论 对于含有高斯和脉冲混合噪声的图像,PGF 滤波是一种很有效的自适应滤波方法。但是该方法在进行同组成

员归类时,引入了 Fisher 线性判别,从而在算法具体实现时需要耗费计算机大量的运算时间,很不适合实时图像处理。本文提出的改进算法是先计算图像的局部能量来判断像素是否为脉冲噪声,若是使用中值滤波滤除,反之将滤波窗口中像素特征变化最小方向上的点归为同组成员,然后可进行 PGF 滤波,可大大提高 PGF 滤波的运算速度。实验结果证明了该算法是一种快速的、能滤除混合噪声又能保护图像边缘信息的滤波方法。

参考文献

- 1 赵荣椿. 数字图像处理导论. 西北工业大学出版社,1995
- 2 Astola J, Haavisto P, Neuvo Y. Vector median filters. In: Proc. of IEEE, vol. 78, April, 1990. 678~689
- 3 Tang K, Astola J, Neuvo Y. Adaptive nonlinear multivariate image filtering for mixed noise removal. In: Proc. of ISCAS, vol. 1. 1993. 427~430
- 4 Deng Y, Kenny C, Moore M S, Manjunath B S. Peer group filtering and perceptual color image quantization. In: Proc. of ISCAS, 1999
- 5 Hewer G, et al. Peer group processing for segmentation: [Technical Report #97.01]. ECE Dept. UCSB, 1997
- 6 Kenny C, et al. Peer Group Image Enhancement. IEEE Tran. ON Image Processing, 2001, 10(2): 326~334
- 7 Cherikh F A, et al. Impulse noise removal in highly corrupted color images. In: Proc. of ICIP, vol. 1. 1996. 997~1000
- 8 Carmona R A, Zhong Sifen. Adaptive Smoothing Respecting Feature Direction. IEEE Tran. ON Image Processing, 1998, 7(3): 353~358
- 9 Di Zenzo S. A note on the gradient of a multi-image. Comput. Vision Graphics Image Proc, 1986, 33: 116~125

(上接第122页)

simistic Pruning 将会用叶节点 L 代替子树 S ; 否则将寻找另一个叶节点对它进行替代, 结果也可能是不存在这样的叶节点。整个剪枝过程如此反复进行, 直至 T 中的所有子树都找不到一个比它更“优秀”的叶节点代替为止。

现在举一个简单的例子说明该方法。如, 某一决策子树 S 覆盖了训练数据集中的 2000 个样本, 该子树有 4 个叶节点, 为了更容易地说明问题, 就用这同样的 2000 个样本作为测试样本集对它进行剪枝。如此, $\Sigma J = 0$, $L(S) = 4$, $N' = 2000$ 。且设该子树被某个叶节点 L 代替后产生 2 个错误 (由于训练数据集也是测试集, 可以容易地观察到), 即 $E = 2$ 。那么有 $SE = \Sigma J + L(S)/2 = 2$, $EE = E + 1/2 = 2.5$, $se(SE) = 1.41$ 。 $EE < SE + se(SE)$, 所以可以用叶节点 L 代替子树 S 。

对决策树进行剪枝的过程无疑是复杂而耗时的, 但它却能够提高最终的准确度, 以及降低树的复杂性, 缩小它的规模, 使知识得以清晰、准确地表现。同时, Pessimistic Pruning 与其它方法比较具有较高的速度。因此我们的剪枝策略采用 Pessimistic Pruning, 以提高完全合成树的精度和表现度。

4 后继工作

数据的类型和数据的分布无疑会影响决策树建立方法的效率, 如何建立它们之间的联系, 我们的新算法在什么样的数据类型及数据分布情况下能获得更优的性能; 算法中的调节因子应该如何根据实际情况设定, 以及如何在分类系统中实时地实现自动设置; 如何提高剪枝的效率。以上是我们下一步在理论研究工作中的主要内容。在实验方面重点是算法的实现和对比。由于 C4.5 已经成为商业系统, 有较多的获取途径, 且具有良好的性能, 各种新方法在形成之初大都做试验与之

比较, 评价新算法的优劣, 因此我们的比较对象为 C4.5。试验与理论是相辅相成, 相互起着促进作用的。

结论 以往的决策树建立方法都是从训练数据集开始执行算法, 在当前数据集中挑选最优属性对数据进行分割, 最终获得决策树。文章提出了一种新的决策树建立方法, 该方法以同类训练数据集作为算法进行操作的对象。以支持度作为衡量属性的标准, 能寻求到确实最支持某一类别的属性, 同时, 在各个同类训练数据集中, 每一个属性的支持度仅仅需要计算一次, 能够提高算法的效率。所有的同类决策树生成后, 对它们的综合并不是无序的, 以性能最好的同类决策树为中心, 其它同类决策树以由优到劣的次序向它靠拢。最后, 对已经形成的完全合成树做剪枝操作, 提高决策树的准确度。

从理论上说, 这种方法高效、精确度高, 有广泛的应用价值。

参考文献

- 1 Apté C, Weiss S. Data Mining with Decision Trees and Decision Rule. Future Generation Computer Systems, 1997, 13: 197~210
- 2 Quinlan J R. Bagging, Boosting, and C4.5. In: Proc. Thirteenth National Conf. on Artificial Intelligence. AAAI Press, 1996. 725~730
- 3 Agrawal R, et al. Database Mining: A Performance Perspective. IEEE Trans. On Knowledge and Data Engineering, 1993, 5(6): 914~925
- 4 Quinlan J R. Simplifying Decision Trees. Int. J. Man-Machine Studies, 1987, 27: 221~234
- 5 Shioya I, Miura T. Knowledge pruning in decision trees. In: Proc. of 12th IEEE Intl. Conf. on Tools with Artificial Intelligence, 2000 (ICTAI). 40~43