

基于改进差分盒算法的织物疵点自动检测方法

曹文梁

(东莞职业技术学院 计算机工程系, 广东 东莞 523808)

摘要: 根据织物检测的实际需要, 提出了基于差分盒算法的改进算法, 在使用 Brodatz 纹理库样本的前提下, 分别在盒子高度确定、盒子总数统计以及网格中盒子数量确定 3 个方面进行改进, 在算法时间和精度两项上, 对经典差分盒算法和改进算法进行了比较, 对比得出了改进算法的时效性; 还应用了改进差分盒算法对 3 种常见纹理的疵点织物进行了计算, 以确认疵点织物, 并验证了改进差分盒算法; 实验通过检出率、误检率、漏检率和检测精度 4 个检测精度参数表明, 改进的差分盒算法可以有效地区分疵点织物和正常织物, 该方法具有很强的实用性。

关键词: 差分盒; 纹理图像; 机器视觉; 分形维数

A Method of Detection of Fabric Defect Based on Improved Differential Boxing—counting Algorithm

Cao Wenliang

(Department of Computer Engineering, Dongguan Polytechnic, Dongguan 523808, China)

Abstract: For the need of detection of fabric defects, the paper presents an improved algorithm based on differential box—counting (ID-BC) method. The algorithm modifies the highness of the boxes, the method of determining the amount of boxes in grids and the statistics of calculating the total of all the boxes on the premise of using Brodatz texture library sample. On the two aspects of the time and accuracy, the paper compared the new algorithm with classical differential box—counting method (DBC) and obtained the efficiency of the improved algorithm. At the same time, the paper calculated defect fabric of the three common texture employing improved differential box—counting method to verify defect fabric. Through four parameters, i. e. detection rate, false drop rate, loss rate and detection precision, the results of experiments demonstrate this new method is effective and can outperform the well—known differential boxing—counting (DBC) method.

Key words: machine vision; fractal dimension; texture image; differential boxing—counting

0 引言

随着科学技术的快速发展与数字图像处理技术日趋成熟, 在纺织品领域中, 使用机器视觉技术进行织物疵点的快速检测技术的研究已成为国际热点问题^[1-2]。目前, 我国检测手段仍停留在人工检测阶段, 检测过程存在着速度慢、精度低、工作强度大等问题^[3-4]。这些问题导致我们越来越不能适应和满足眼前经济大发展前提下的大规模的工业生产的要求。为此, 将机器视觉技术应用到织物疵点检测中即可解决上述问题, 同时大幅度的降低了检测成本, 提高了检测效率。

现实中, 物体轮廓特征是复杂的, 它呈现出许多复杂的不规则自然形态。分形维数的描述可以为这些不规则的形态给出确定的数学模型。因此, 将分形理论应用于织物疵点检测, 通过求取和对比正常织物图像和待测织物图像的分形维数来判断待测织物是否存在疵点是一个可行的检测方法。根据对应情况的不同, 计算分形维数有很多种不同的方法^[5-7]。数盒子法由于算法实现简单, 并具有自动计算性的特点而得到了广泛的应用, 其下的差分盒方法在高效和实用性上显得更为突出, 因此差分盒方法被公认为最高效实用的计算分形维数的方法。

经典差分盒算法主要采用的是二维图像处理技术^[8]。由于提取分形维数时计算量较大, 经典差分盒算法并不能完全适用于疵点检测处理要求, 缺乏实时性。盒子数量计算的不准确和盒子模型测定的不精确也是经典差分盒方法本身一直存在的问题^[9]。针对上述问题, 本文提出改进差分盒算法的一种应用于检测织物疵点。完全不同于经典差分盒算法的盒子数量统计和盒子高度选取方法, 并且通过纹理库图像样本, 证明该方法在算法时间和拟合误差两方面, 较于经典差分盒算法, 都更加优秀。

1 差分盒算法

差分盒算法应用自相似理论, 我们可以将 $M * M$ 的灰度图像想象为一个三维坐标系中的立体图形, 其中的图像的长、宽分别为坐标轴的 x, y 轴, 对应每一点 (x, y) 的灰度值为坐标 z 。将分辨率为 $M * M$ 的图像分解为 $r * r$ 的网格, r 的取值范围是 $2 \leq r \leq M/2$, 且必须为整数, 将每个网格中堆放 $r * r * h$ 的盒子, 盒子高度 h 的计算方法为:

$$h = r * G / M \tag{1}$$

其中: G 为图像的灰度级, 由此可知每个网格中, 盒子的数量 n_r 为:

$$n_r = k - l + 1 \tag{2}$$

其中: k, l 分别代表灰度最大值所在盒子数和灰度最小值所在的盒子编号, 则在网格 r 下, 全部盒子的数量 N_r 为:

$$N_r = \sum n_r \tag{3}$$

收稿日期: 2014-01-12; 修回日期: 2014-03-05。

基金项目: 国家自然科学基金项目(70971043)。

作者简介: 曹文梁(1980-), 男, 江西泰和人, 硕士, 主要从事算法设计与分析、网络安全等研究。

由不同的网格长度 r ，并对 $\log(N_r)$ 和 $\log(1/r)$ 做最小二乘拟合，可得图像的分形维数 D 为：

$$D = \lim_{r \rightarrow 0} \frac{\log(N_r)}{\log(1/r)} \quad (4)$$

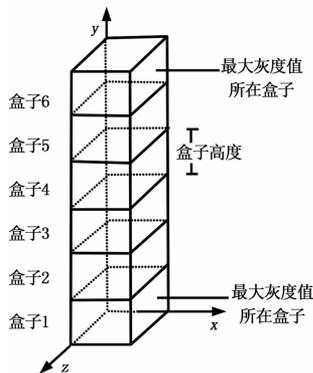


图 1 差分盒算法盒子模型示意图

经典差分盒算法主要存在以下几个问题：

1) 盒子高度选取问题：在 $r * r * h$ 的盒子中， r 为网格宽度，为固定值，而主要的变化为盒子高度 h ，由式 (1) 可知，盒子高度 h 仅取决于 r ，而与图像灰度值分布无关，这样，当两幅 $M * M$ 图像同时取 $r * r$ 的网格时，如果两幅图像灰度变化差距较大时，使用相同的盒子模型便不能取得精准的分形维数。因此需要根据图像灰度级的变化调整盒子高度。又因为更小的测量标准一般可得到更为准确的结果，因此新的盒子高度 $h' < h$ 。

2) 网格盒子数量计算问题：在经典差分盒算法中，求取网格中盒子数量时，使用的是盒子数量差值，这样，当最大灰度值和最小灰度值差距相同，而所在盒子位置不同时，如图 2 所示，可能会出现不同的结果，因此使用灰度级差来计算盒子数量，更为合理。

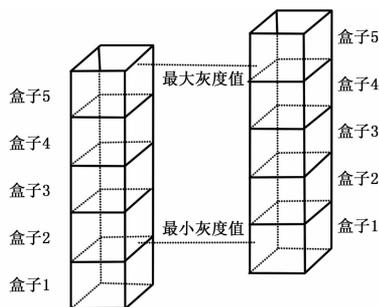


图 2 灰度值大小对网格盒子数的影响

3) 重复计算问题：在差分盒算法中，需要反复求取分形维数，造成计算时间过长，当图像分辨率较大时，对算法时间影响较大。

2 改进差分盒算法

2.1 算法的改进思想

根据上一节提出的差分盒算法存在的问题，现进行 3 个方面的改进：

(1) 引入新的盒子高度定义方法，新的盒子高度 $h' = h/c$ 其中， c 是一个大于 1 的常数，如图 3 所示，因为 $h' < h$ ，使

每行网格中的总的盒子数量更多，从而得到更为准确的计算结果；

(2) 提出新的基于灰度值差距的盒子数量计算方法，解决特定灰度值对盒子数量产生的影响。如图 2 所示，左右两个网格大小与盒子高度均相同，最大灰度值与最小灰度值之差也相同，但由于最大灰度值和最小灰度值的具体数值不同，造成网格中的盒子个数不同。

(3) 同时计算相同大小的几个网格中的盒子数量，减少重复计算时间。

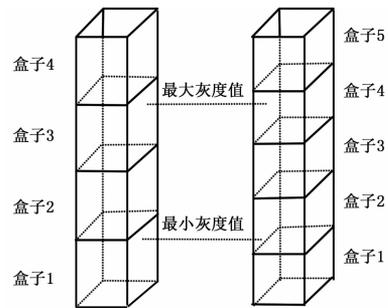


图 3 盒子高度对计算精度的影响

2.2 基于盒子高度的改进

新的盒子高度定义方法为：

$$h' = \frac{h}{1 + 4\sigma} \quad (5)$$

其中： h 为经典差分盒算法中盒子高度的定义方法， σ 为图像的灰度方差，对比经典差分盒算法，改进算法中的盒子高度更小，例如，一副 $256 * 256$ 的灰度图像，其灰度方差为 15，由式 (5) 可知，改进算法中盒子高度 $h' = h/61$ ，因此新的算法根据图像灰度调整盒子高度，得到更为精确的覆盖原灰度图像的盒子个数。

2.3 基于网格盒子数量计算方法的改进

设某一网格中的最大灰度值和最小灰度值分别为 g_{\max} 、 g_{\min} ，则网格中的盒子个数 n_r 的计算方法为：

$$n_r = \text{ceil}((g_{\max} - g_{\min})/h') \quad (6)$$

其中： ceil 为取大于括号内数的最小整数， h' 为对应的盒子高度，本改进方法直接使用灰度值的差值求取盒子个数，相比经典差分盒算法，更为精确，减少了灰度值对盒子数量造成的影响。

2.4 基于总体盒子数量统计方法的改进

本改进方法改变了经典差分盒算法中，网格中盒子数量的累加顺序，提高计算效率，具体计算过程的伪代码如下：

```

r = M/s
for j=0 to s-1
  for k=0 to s-1
    for H=0 to (X+M-1-k)/s-1
      for L=0 to (Y+M-1-j)/s-1
        MAX = max(F[i, j+L*s; j+L*s+s-1, k+H*s; H+H*s+s-1])
        MIN = min(F[i, j+L*s; j+L*s+s-1, k+H*s; H+H*s+s-1])
        G[H, L] = MAX - MIN + 1;
      endfor
    endfor
  endfor

```

```

for H=0 to (X-1-k)/s-1
for L=0 to (Y-1-k)/s-1
Nr=sum(G[L:L+r-1,H:H+r-1])
F=log10(Nr)/log(r)
endfor
endfor
endfor
endfor

```

其中：X、Y 代表图像的长、宽，M 代表窗口分辨率，s 代表网格分辨率，MAX、MIN 分别代表网格中的最大灰度值和最小灰度值。

3 实验及结论

由于分形维数基于自相似理论得出，是一个相对值，因此对于任何图像，均没有绝对正确的分形维数，而在分形维数计算时，最后一步拟合误差就是评定分形维数求取是否准确的最重要标准，拟合误差的计算公式为：

$$y_r = ax + b \quad (7)$$

$$FE = \frac{1}{n} \sum_{r=1}^n \frac{|\log N_r - y_r|}{1 + a^2} \quad (8)$$

其中：y_r 为拟合直线的纵坐标，x 为横坐标；a 为直线的斜率，b 为截距；FE 为拟合误差 (Fit Error)，logN_r 为差分盒算法得出的直线纵坐标。

在待测样本的选取上，为了体现普遍性原则，本文选用中的灰度图像作为检测样本。Brodatz 纹理库是目前在全世界范围内，认同度最高的图像纹理测试标准图像，该纹理库包含了各种常见的灰度纹理图像，本文选择的 8 副纹理库中的图像如图 4 所示。

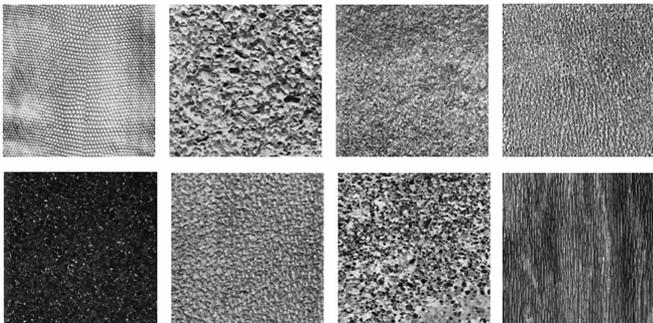


图 4 纹理库图像样本

表 1 IDBC 算法与 DBC 算法拟合误差对比

图像编号	DBC		IDBC	
	D	FE	D	FE
D03	2.73	0.025	2.86	0.007
D05	2.65	0.021	2.65	0.010
D09	2.73	0.022	0.77	0.006
D24	2.86	0.011	0.77	0.005
D33	0.61	0.017	2.49	0.007
D54	2.75	0.017	2.74	0.007
D68	2.74	0.019	2.75	0.005
D92	2.83	0.006	2.60	0.005

另外，因为织物疵点检测的实时性要求，算法时间也是评价该算法是够有效的重要因素之一。DBC 算法在求取一次网格中盒子个数的过程中需要进行 4 次加运算、2 次减运算、2 次乘运算和 2 次除运算，以及 2s² 次灰度比较。在进行盒子数量累加时还需要两次加运算和一次减运算。算法总计循环 (M/s)² 次，因此平均算法复杂度为：6(M/s)² 次加运算，3(M/s)² 次减运算，2(M/s)² 次乘运算、2(M/s)² 运算以及 2M² 次比较运算。IDBC 算法中，在统计最大最小灰度值的过程中，比 DBC 算法少一次加运算，使用 sum 函数进行统计时需要增加 (M/s)² 次加运算，遍历相关区域时有需要额外的 2 次加运算和 2 次减运算。算法总计循环次数为 xy/s²，并需要 2xy 次比较运算。因此算法平均复杂度为：7+(M/s)² 次加运算，5 次减运算，2 次乘运算和 2 次除运算。IDBC 与 DBC 算法的平均算法复杂度对比如表 2 所示。

表 2 IDBC 与 DBC 算法复杂度对比

运算方式	IDBC	DBC
加运算	7+(M/s) ²	6(M/s) ²
减运算	5	3(M/s) ²
乘运算	2	2(M/s) ²
除运算	2	2(M/s) ²
比较运算	2s ²	2M ²

对应表 1 选取的 8 幅纹理库图像的两种算法的运行时间对比如图 5 所示。

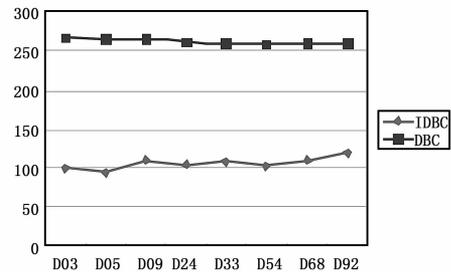


图 5 IDBC 算法与 DBC 算法运行时间对比

为了验证改进算法应用于织物疵点检测的实际效果，本文选择了几种典型纹理织物的疵点图像进行检测。如图 6 所示。

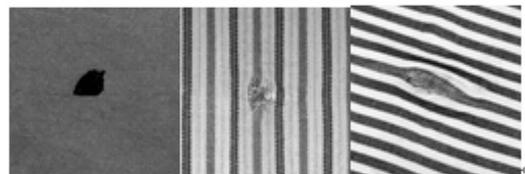


图 6 典型织物疵点图像

为验证结果的正确性，本文采用检出率、误检率、漏检率和检测精度 4 个常用检测系统精度参数：

$$\text{检出率} = \frac{\text{检测出的疵点个数}}{\text{疵点总数}} \times 100\% \quad (9)$$

$$\text{误检率} = \frac{\text{误报疵点个数}}{\text{样本总数}} \times 100\% \quad (10)$$

$$\text{漏检率} = \frac{\text{未检测出的疵点个数}}{\text{样本总数}} \times 100\% \quad (11)$$

