

纹理分割的最小二乘滤波器

陈贺新 戴逸松 付 平 钱神恩

(吉林工业大学)

摘要: 本文根据纹理结构的预知性, 推导出了一种新的纹理分割方法——最小二乘滤波器。该方法的特点是所用滤波器少(图像中需要分析几种纹理, 就对应几个滤波器), 而且每一个滤波器必定增强对应的纹理区域, 这对最后纹理分割区域带来了极大的方便。再者该方法节省内存、计算速度快。

一、引 言

到目前为止, 国内外研究出来的纹理分析方法都是在纹理结构情况未知条件下进行分析的。例如, Frei 和 Chen 在1977年提出的九个滤波掩码^{[1][2]}; Laws 的 3×3 维和 5×5 维的滤波器法^[3]; Ade 于1983年提出的特征滤波器^[4]等。这三种滤波器是近年来纹理分析领域中具有代表性的滤波器。由于它们都未利用纹理结构的预知性, 因此在纹理分析中, 这些方法都具有一定的“盲目性”。实际中, 我们对于所要分析或分割的各种纹理, 都具有一定的预先了解。例如, 对于遥感图像而言, 森林在图像呈现的纹理结构大致应该如何是应该预先有所了解的。否则, 即使森林纹理区域在图像中用上述“三种滤波器分割出来了, 人们也未必知道该区域就是森林区域。又如, 在工业产品表面质量检验中, 无损的表面纹理也是预知的。由此可见, 充分利用纹理的预知性进行纹理分割是非常实际的。本文就是基于纹理的预知性推导出最小二乘滤波器的。

二、最小二乘滤波器

我们假定原始图像为 $F(x, y)$, $1 \leq x \leq N$ 及 $1 \leq y \leq M$, 由 r 种纹理 $f_1(x_1, y_1), f_2(x_2, y_2), \dots, f_r(x_r, y_r)$ 组成。其中 $x, y \in \Omega, x_i, y_i \in \Omega_i (i = 1, 2, \dots, r)$ 。 Ω 是 $F(x, y)$ 的定义域, Ω_i 是 $f_i(x_i, y_i)$ 的定义域 ($i = 1, 2, \dots, r$), 二者之间的关系可写成:

$$\Omega = \sum_{i=1}^r \Omega_i \quad \text{且 } \forall i, j \text{ 及 } i \neq j$$

$$\Omega_i \cap \Omega_j = \phi \quad \Omega_i \in \Omega (i, j = 1, 2, \dots, r) \quad (1)$$

相似地, $F(x, y)$ 和每一个 $f_i(x_i, y_i)$ 可以分别定义为总集合与子集合, 而且可用下式表示二者间的关系:

$$F(x, y) = \sum_{i=1}^r f_i(x_i, y_i) \quad \text{且 } \forall i, j \text{ 及 } i \neq j$$

$$f_i(x_i, y_i) \cap f_j(x_j, y_j) = \phi$$

$$f_i(x_i, y_i) \in F(x, y) \quad (i, j = 1, 2, \dots, r) \quad (2)$$

在选择工作窗口尺寸为 $n \times n$ ($n \ll N, M$) 的情况下, 我们的目的是推导出—组 (包括 r 个) 滤波器 $h_i(k, l)$, $i = 1, 2, \dots, r$, $1 \leq k, l \leq n$, 每个滤波器 $h_i(k, l)$ 对应一种纹理有效 (即, 对应该种纹理, 在卷积输出时呈现极大值)。

设以第 i 种纹理构成的满幅图像为 $F_i(x, y)$ (称作基底纹理图像), $i = 1, 2, \dots, r$, (注: $x, y \in \Omega, f_i(x_i, y_i) \in F_i(x, y)$)。为了求得对应第 i 种纹理有效的滤波器 $h_i(k, l)$, 需对下式相对于 $h_i(k, l)$ 各值求最小值:

$$\forall i, \{\psi = \sum_{x, y} [\sum_{k=1}^n \sum_{l=1}^n h_i(k, l) * F(x+k-1, y+l-1) - F_i(x, y)]^2\} \\ i = 1, 2, \dots, r \quad (3)$$

为了推导方便, 将滤波器 $h_i(k, l)$ ($n \times n$ 维) 按“行优先”规则改写成矢量形式—滤波器矢量 H_i , 即

$$\vec{H}_{i, LSF} = [h_i(1, 1), h_i(1, 2), \dots, h_i(1, n), h_i(2, 1), h_i(2, 2), \\ \dots, h_i(2, n), \dots, h_i(n, 1), h_i(n, 2), \dots, h_i(n, n)]^t \quad (4)$$

同样地, 将 $n \times n$ 窗口内 (对应 $F(x, y)$ 像素点) 的原图像 $n \times n$ 维像素组写成原图像窗内矢量 \vec{F}_w , 即

$$\vec{F}_w = [F(x, y), F(x, y+1), \dots, F(x, y+n-1), F(x+1, y), \\ F(x+1, y+1), \dots, F(x+1, y+n-1), \dots, F(x+n-1, y), \\ F(x+n-1, y+1), \dots, F(x+n-1, y+n-1)]^t \quad (5)$$

上述矢量的符号“LSF”表示最小二乘滤波器; “W”表示窗口内; “t”表示转置。于是, 式 (3) 可重新写成矢量形式如下

$$\forall i, \{\phi = \sum_{x, y} (\vec{H}_{i, LSF}^t \vec{F}_w - F_i(x, y))^2\} \\ = \sum_{x, y} \phi_i(x, y) \quad i = 1, 2, \dots, r \quad (6)$$

相对应滤波器 $h_i(k, l)$ 各元素求式 (3) 与式 (6) 的最小值, 即

$$\left. \begin{aligned} \sum_{x, y} \frac{\partial \phi_i(x, y)}{\partial h_i(1, 1)} &= 0 \\ \sum_{x, y} \frac{\partial \phi_i(x, y)}{\partial h_i(1, 2)} &= 0 \\ &\vdots \\ \sum_{x, y} \frac{\partial \phi_i(x, y)}{\partial h_i(1, n)} &= 0 \\ \sum_{x, y} \frac{\partial \phi_i(x, y)}{\partial h_i(n, 1)} &= 0 \\ \sum_{x, y} \frac{\partial \phi_i(x, y)}{\partial h_i(n, 2)} &= 0 \\ &\vdots \\ \sum_{x, y} \frac{\partial \phi_i(x, y)}{\partial h_i(n, n)} &= 0 \end{aligned} \right\} \quad (7)$$

经计算整理后, 可得最小二乘滤波器矢量 $\vec{H}_{i,LSF}$ 为

$$\vec{H}_{i,LSF} = \vec{A}^{-1} \vec{G} \quad (8)$$

式 (8) 中的矩阵 \vec{A} 叫做窗内原始图像的自相关矩阵, 且由下式表示:

$$\vec{A} = \sum_{x,y} \begin{pmatrix} F^2(x,y) & F(x,y)F(x,y+1) \cdots F(x,y)F(x+n-1,y+n-1) \\ F(x,y)F(x,y+1) & F^2(x,y+1) \cdots F(x,y+1)F(x+n-1,y+n-1) \\ \cdots & \cdots & \cdots & \cdots \\ F(x,y)F(x+n-1,y+n-1) & F(x,y+1)F(x+n-1,y+n-1) \cdots \\ F^2(x+n-1,y+n-1) & \cdots & \cdots & \cdots \end{pmatrix} \\ = \sum_{x,y} \vec{F}_w \cdot \vec{F}_w^t \quad (9)$$

式 (8) 中的 \vec{G} 矢量是原始图像 $F(x,y)$ 与已知第 i 种基底纹理图像 $F_i(x,y)$ 之间的互相关矢量, 且为

$$\vec{G} = \sum_{x,y} \begin{pmatrix} F(x,y)F_i(x,y) \\ F(x,y+1)F_i(x,y) \\ \vdots \\ F(x,y+n-1)F_i(x,y) \\ \vdots \\ F(x+n-1,y)F_i(x,y) \\ F(x+n-1,y+1)F_i(x,y) \\ \vdots \\ F(x+n-1,y+n-1)F_i(x,y) \end{pmatrix} = \sum_{x,y} \vec{F}_w \cdot F_i(x,y) \quad (10)$$

式 (8) 中 $\vec{H}_{i,LSF}$ 矢量各元素按“行优先”反逆排列 (见式(4)) 后, 即可得到对应原图像中第 i 种纹理有效的最小二乘滤波器 $h_i(k,l)$, 如下式表示:

$$[h_i(k,l)] = \begin{pmatrix} h_i(1,1) & h_i(1,2) & \cdots & h_i(1,n) \\ h_i(2,1) & h_i(2,2) & \cdots & h_i(2,n) \\ \cdots & \cdots & \cdots & \cdots \\ h_i(n,1) & h_i(n,2) & \cdots & h_i(n,n) \end{pmatrix} \quad (11)$$

显然, 最小二乘滤波器的求得需要 r 种纹理的全幅图像 (也叫基底纹理图像) $F_i(x,y)$, $i = 1, 2, \dots, r$, 以及原始图像 $F(x,y)$, $1 \leq x \leq N, 1 \leq y \leq M$ 。通过计算 $F(x,y)$ 与 $F_i(x,y)$ 的互相关性, 以及 $F(x,y)$ 本身的自相关性, 就可以求得每一种纹理对应有效的最小二乘滤波器 $h_i(k,l)$ 。很清楚, 对于含有 r 种纹理的原始图像来说, 应该求得 r 个最小二乘滤波器。

在最小二乘滤波器法中, 每一种纹理对应一个最小二乘滤波器, 该滤波器表征着对应纹理的唯一特征性, 并不取该决策纹理的区域性与其边界 (见式(6))。每一种纹理有效的最小二乘滤波器与原始图像卷积时, 将会突出对应纹理的区域。不过, 正如前面提到的, 这种最小二乘滤波器法需要预先对纹理有所了解, 以便获某一种纹理的基底纹理图像 $F_i(x,y)$ 。值得指出, 有关 $F_i(x,y)$ 的预知性与共生知识并不意味着预先知道有关纹理区域与边界的特殊信息。

三、纹理区域分割及实验

对于 $N \times M$ 的原始纹理图像 $F(x, y)$ 和最小二乘滤波器 $h_i(k, l)$ 来说, 以 $n \times n$ 工作窗口的响应图像 $R_i(x, y)$ 由下式定义:

$$\forall x, y, R_i(x, y) = \sum_{k=1}^n \sum_{l=1}^n F(x+k, y+l) h_i(k, l) \quad (12)$$

通常, 滤波输出以能量形式出现^{[2][3]}, 且由下式定义:

$$\forall x, y, R_{E_i}(x, y) = [R_i(x, y) - \bar{R}_i]^2 \quad (13)$$

式 (12) 中 \bar{R}_i 表示第 i 种纹理有效最小二乘滤波器与原始图像卷积输出的平均灰度值 (或称为直流照度):

$$\bar{R}_i = \frac{1}{NM} \sum_{x, y} R_i(x, y)$$

这里 NM 为图像幅尺寸。

图 1 按 Julesz 以及 Caelli 等人提出的理论纹理^{[5][6]} 给出的原始图像 (图 1(a) 和图 1(b) 中标有“Input”字样的图像), 以及各自对应的两幅能量输出图像 (图 1(a) 与图 1(b) 中标有 LSF 字样的图像)。对应的最小二乘滤波器如表 1 所示。

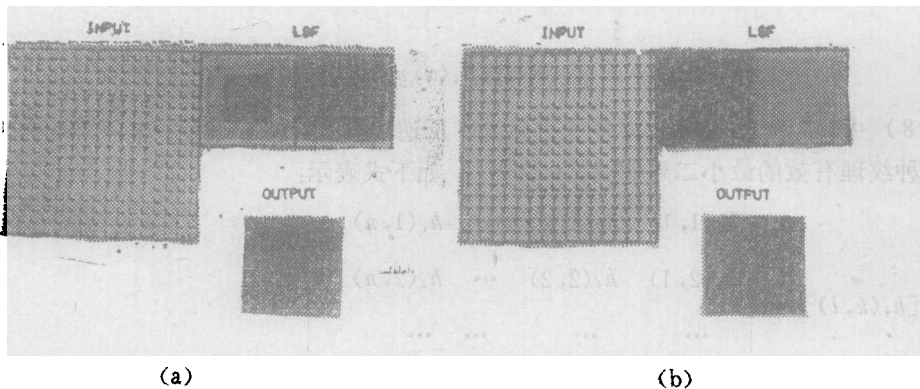


图 1 两幅理论纹理实验结果 ($N = M = 128$)

表 1 对应图 1 图像的两组最小二乘滤波器

图 1 (a)	图 1 (b)
最小二乘滤波器 1 (中间纹理)	最小二乘滤波器 1 (中间纹理)
$\begin{pmatrix} 1.00 & 0.04 & 0.87 \\ 0.04 & -0.05 & 0.04 \\ 0.74 & 0.04 & 0.79 \end{pmatrix}$	$\begin{pmatrix} 1.00 & 0.51 & 0.36 \\ 0.39 & 0.20 & 0.13 \\ 0.36 & 0.19 & 0.15 \end{pmatrix}$
最小二乘滤波器 2 (外部纹理)	最小二乘滤波器 2 (外部纹理)
$\begin{pmatrix} 0.60 & 0.03 & 0.85 \\ 0.03 & -0.05 & 0.03 \\ 1.00 & 0.03 & 0.56 \end{pmatrix}$	$\begin{pmatrix} 0.23 & 0.27 & 0.23 \\ 0.77 & 0.68 & 0.32 \\ 1.00 & 0.42 & 0.17 \end{pmatrix}$

图 2 给出了一幅实际由泡纸板和地毯构成的图像 (图 2 中标“Input”字样), 以及对应的两幅能量输出图像 (图 2 中标有“LSF”字样)。对应的最小二乘滤波器如表 2 所示。值得指出, 这里 (图 1 和图 2) 的能量输出图像都是按式 (12) 算得后依纹理图案尺寸大小采取平均处理后的结果。

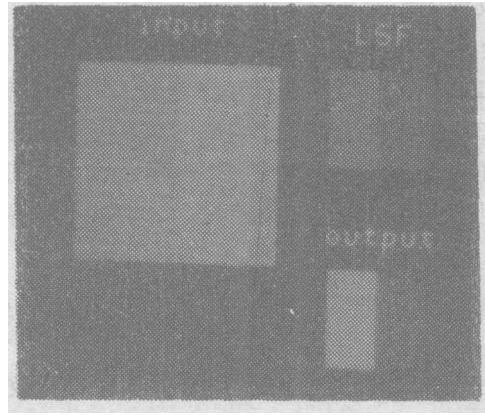


图 2 一幅实际的泡纸板与地毯图像实验结果
($N = M = 128$)

纹理的最后分割是利用每一对应象素 (各自能量输出图像上) 的投影 (即互相关性) 完成的。通常, 有两种方法进行图像纹理滤波后的分割区域。一种是确定纹理区域的边界, 而后再进行分割区域; 另一种基于能量输出投影的相似性与差别性来分割区域。二者都是利用滤波后能量输出的投影矢量完成。在此, 我们采用第二种方法, 并定义一个标量积^[1]的概念。

表 2 对应图 2 图像的最小二乘滤波器

最小二乘滤波器 1 (左)			最小二乘滤波器 2 (右)		
0.93	-0.08	1.00	0.13	0.45	-0.69
0.61	0.21	0.07	-0.76	1.00	0.31
0.99	0.50	-0.74	0.08	0.90	0.09

$$l(x, y) = \vec{R}_E(x, y) \vec{R}_E^*(x^*, y^*) \quad (14)$$

其中, $\vec{R}_E(x, y)$ 表示在 (x, y) 位置上的能量矢量 (即投影矢量) ($R_{E1}(x, y), R_{E2}(x, y), \dots, R_{E\tau}(x, y)$), 而 $\vec{R}_E^*(x^*, y^*)$ 表示在随机选择的位置 (x^*, y^*) 上能量矢量 ($R_{E1}^*(x^*, y^*), R_{E2}^*(x^*, y^*), \dots, R_{E\tau}^*(x^*, y^*)$)。这里 τ 意味着原始纹理图像中包括 τ 种纹理, 显然, 卷积能量输出图像共有 τ 幅。上述纹理必须选择适当的值范围, 其原因是纹理的彼邻性与规律性都与选择处理的值范围有关。图 1 (a) 和 (b), 以及图 2 中标有“Output”字样的图像就是利用式 (14) 最后分割的结果。

四、最小二乘滤波器法分割纹理的实验步骤

如图 3 所示, 利用最小二乘滤波器法分割纹理的系统输入是两幅数字图像, 一幅为基

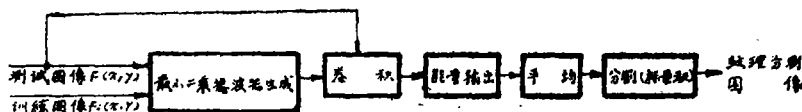


图 3 最小二乘滤波器法分割纹理系统框图

底纹理图像 (在图 3 中为训练图像) $F_i(x, y)$, 另一幅是原始图像 (在图 3 中为测试图像) $F(x, y)$ 。在最小二乘滤波器生成中, 对应 r 种纹理有效的 r 个最小二乘滤波器由式 (6)~(11) 算得。每种纹理有效最小二乘滤波器与原始图像卷积利用式 (12) 之后, 再应用式 (13) 将其转换成能量输出图像。以子图案尺寸为近似基础尺寸, 对能量输出图像进行平均处理。最后, 依式 (14), 纹理区域分割图像输出。如果原始图像中含有 r 种纹理, 就会生成对应的 r 个最小二乘滤波器, 也就会获得 r 幅能量输出图像 (或称分类集)。

五、结 论

最小二乘滤波器法适用于具有预知性纹理的分割与分类。实际中, 大多数纹理, 无论是工业产品表面纹理还是其它实际图像上的纹理, 都具有预知性, 上述基于二阶纹理统计特性纹理分析的结果表明, 最小二乘滤波器对于纹理区域分割非常有效。

这种自适应滤波技术最重要一个特点是它能最佳提取对应纹理的特征。这为最终的有效分割纹理区域提供了保证。

本工作曾与应用光学开放实验室的有关专家商讨研究, 并得到多方面的支持与协助。

参 考 文 献

- [1] Frei, W. and Chen, C., IEEE Trans., C-26, 1977, 988—998
- [2] Caelli, T.M., Spatial Vision, 1, No. 1, 1985, 19—30
- [3] Laws, K., SPIE, 238, 1980, 376—380
- [4] Ade, F., Signal Processing, 5, 1983, 451—457
- [5] Julesz, B., Nature, 290, 1984, 91—97
- [6] Caelli, T.M., Biological Cybernetics, 39, 1980, 21—26

Least Square Filters for the Texture Segmentation

Chen Hexin Dai Yisong Fu Ping Qian Shen

Abstract

Based on the foreknowledge of the textures in a given image a new method for the texture segmentation---Least Square Filters are derived in this paper. The method uses a little filter, the number of which depends on the number of the textures needed to be analyzed, and each filter enhances, certainly, the texture region related. All of these are convenient for the texture segmentation. The method uses a little memory space and its computation speed is fast.