

# 基于向量机的图像插值算法研究

郑胜<sup>1),2)</sup> 田金文<sup>2)</sup> 柳健<sup>2)</sup>

<sup>1)</sup>(三峡大学,宜昌 443002) <sup>2)</sup>(华中科技大学图像信息处理与智能控制教育部重点实验室,武汉 430074)

**摘要** 为了更好地进行图像插值,可首先利用最小二乘向量机对原始图像的局部区域进行灰度曲面最佳拟合,然后在拟合曲面上进行未知像素点的灰度值估计,同时提出了运用测试图像进行参数优化的方法,并以径向基核函数为例导出了区域图像的插值系数矩阵,进行了图像放大插值实验验证。实验结果表明,基于支持向量机的图像插值算法具有很强的适应性,其性能与 Cubic 技术相当,但效率更高。

**关键词** 图像插值 最小二乘向量机 插值系数矩阵 参数优化

中图分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2005)03-0338-06

## Research of SVM-based Image Interpolation Algorithm Optimization

ZHENG Sheng<sup>1),2)</sup>, TIAN Jin-wen<sup>2)</sup>, LIU Jian<sup>2)</sup>

<sup>1)</sup>(China Three Gorges University, Yichang 443002)

<sup>2)</sup>(State Education Commission Key Laboratory for Image Processing and Intelligent Control, Huazhong University of Science and Technology, Wuhan 430074)

**Abstract** The image intensity surface for the local neighborhood of image is well-fitted by the least squares support vector machine (lssvm), and then the gray level interpolation is implemented on the fitted intensity surface. The interpolation coefficient matrix of the local neighborhood is deduced from the lssvm with the radial basis function (rbf) kernel function, as an example. A method using the interpolation evaluating merit figure, psnr, to optimize the svm parameters is proposed. With the selected parameters, the computer interpolation experiments are carried out. The experimental results demonstrate the svm-based interpolation algorithm has similar performance to cubic one but providing higher efficiency.

**Keywords** image interpolation, least squares support vector machine (LSSVM), interpolation coefficient matrix, parameter optimization.

## 1 引言

信息的采样和插值是信息处理领域中重要的技术,其在许多方面有着广泛的应用<sup>[1]</sup>。大家知道,插值是一个利用离散样本来估计连续事件的中间值的过程<sup>[2]</sup>,其已被广泛地用于对数字图像的放大、缩小等处理。通常,图像的放大是一个图像与一个核的卷积过程,目前已有多种内插算法<sup>[3]</sup>,而且这些算法最终都将内插问题归结为一个函数约束极小值的优化过程<sup>[4]</sup>。一般来讲,大多数图像都具有较

强的空间相关性,即相邻的像素点往往具有相似的图像灰度值,因此,就可以把图像看作曲面,并用曲面最佳拟合的方法先把原始图像用拟合曲面来代替,然后再在拟合曲面上作未知像素点的灰度值估计。图像的采集就是对这个空间曲面进行采样的过程,其中的每一个像素就是一个采样点,而图像的内插就是利用这些样本点的灰度值去估计与之邻近的未知像素的灰度值。曲面插值算法的关键是如何得到最佳的拟合曲面,以使用其来准确地估计曲面上的灰度值。

近年来,基于统计学习理论的支持向量机

基金项目:教育部博士点基金资助项目(20010487030)

收稿日期:2004-02-12;改回日期:2004-10-08

第一作者简介:郑胜(1965 - ),男,副教授。1992年获华中师范大学无线电电子学专业硕士学位,现为华中科技大学博士研究生。主要从事图像处理、目标检测与识别、神经网络与智能控制等方面的研究工作。E-mail: zhengsheng6511@sohu.com

(support vector machine, SVM), 作为新的数据分类和函数估计工具, 得到了巨大的发展<sup>[5]</sup>。它已经在模式识别、目标检测、函数估计等方面得到了广泛的应用。特别是一种改进的 SVM, 即最小二乘支持向量机 (least squares SVM, LSSVM)<sup>[6]</sup> 的提出, 其与标准向量机的解最终都归结为凸二次函数的寻优问题, 所不同的是 LSSVM 的求解是通过求解一组线性方程来完成的。由于它具有比标准向量机低得多的计算复杂度和内存需求, 从而为其能在大数据量的图像处理中得到应用提供了可能。本文讨论了基于径向基核函数 (radial basis function, RBF) 的映射 LSSVM 图像灰度插值技术和参数优化方法, 并进行了图像放大插值的试验验证。结果表明, 本文的方法有效。

## 2 SVM 插值理论和方法

### 2.1 最小二乘向量机

给定训练样本集  $\{(x_i, y_i), i = 1, \dots, N\}$ ,  $x_i \in \mathbf{R}^n, y_i \in \mathbf{R}$ , 其中  $x_i$  是第  $i$  个具有已知输出数据  $y_i$  的输入矢量。在 LSSVM 中, 应考虑以下的优化问题, 即

$$\min_{w, b, e} J(w, b, e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^N e_k^2 \quad (1)$$

满足

$$y_k [w^T \varphi(x_k) + b] = 1 - e_k, k = 1, \dots, N$$

式中  $\varphi: \mathbf{R}^n \rightarrow \mathbf{R}^m$  是特征映射函数, 用于实现从原空间到高能特征空间的映射;  $w \in \mathbf{R}^m$  为高能空间中的加权矢量,  $e_k \in \mathbf{R}$  为误差项,  $b \in \mathbf{R}$  为偏移量; 损失函数  $J$  由方差和与调整项组成, 这些调整项的重要性由常数  $\gamma$  确定。在有噪声的情况下, 应取较小的  $\gamma$  值, 以避免过学习的情况。正如文献<sup>[6]</sup>中所述, 式(1)的解可归结为求解如下形式的线性方程组:

$$\begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \mathbf{Z} + \gamma^{-1} \mathbf{I} \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{Y} \end{bmatrix} \quad (2)$$

其中,  $\mathbf{Y} = [y_1 \dots y_N]^T, \mathbf{1} = [1 \dots 1]^T, \alpha = [\alpha_1 \dots \alpha_N]^T, \mathbf{Z}_{k,l} = \varphi(x_k)^T \varphi(x_l); k, l = 1 \dots N$ , 而根据 Mercer 条件则存在一个映射函数  $\varphi$ , 并可选择一个核函数  $K$  满足下列条件

$$K(x_k, x_l) = \varphi(x_k)^T \varphi(x_l); k, l = 1, \dots, N \quad (3)$$

这样基于 LSSVM 的函数估计表达式为

$$y(x) = \sum_{k=1}^N \alpha_k K(x, x_k) + b \quad (4)$$

式中, 参数  $\alpha_k$  和  $b$  为式(2), 式(3)的解。式(4)的许多特性均取决于满足 Mercer 条件的核函数  $K$ , 其种类很多, 目前以径向基核函数 (RBF)  $K(x, x_k) = \exp\left\{-\frac{|x-x_k|^2}{\sigma^2}\right\}$  使用最广。

### 2.2 映射 LSSVM

在图像处理中, 典型的图像像素邻域的大小为  $3 \times 3$  到  $9 \times 9$ , 很明显, 在一个像素的邻域中, LSSVM 的输入矢量由像素坐标定义, 而输出目标则为像素的灰度值。对任何一个像素  $(i, j)$  的邻域, 其输入矢量通常可以用  $\{(i + \delta_i, j + \delta_j); |\delta_i| \leq m, |\delta_j| \leq n\}$  形式来描述。如果从所有输入矢量中减去  $(i, j)$ , 则该输入矢量集就可以转换成相同的矢量集  $\{(\delta_i, \delta_j); |\delta_i| \leq m, |\delta_j| \leq n\}$ , 其中  $m$  和  $n$  分别为像素邻域的水平 and 垂直像素数的一半。若将矢量集  $\{(\delta_i, \delta_j); |\delta_i| \leq m, |\delta_j| \leq n\}$  的所有矢量都加上  $(i, j)$ , 则可恢复为原来的输入矢量集。采用这种映射技术, 即可将图像处理中的 LSSVM 的学习问题转换为一个在输入矢量集相同但输出目标不同的矢量空间中的问题。这样当像素邻域的大小确定后, 则输入矢量集  $\{(\delta_i, \delta_j); |\delta_i| \leq m, |\delta_j| \leq n\}$  就构成一个常矢量空间。

注意到求解式(2)时, 实际上只需对如下  $N \times N$  矩阵求逆即可

$$\Omega = K(x_i, x_j) + \gamma^{-1} \mathbf{I} \quad (5)$$

其中,  $N = (2m + 1)(2n + 1)$  为样本数。事实上, 式(3)中的第 2 个方程式已给出了  $\mathbf{1}b + \Omega\alpha = \mathbf{Y}$ , 而加上第 1 个方程式则得其解为

$$b = \frac{\mathbf{1}^T \Omega^{-1} \mathbf{Y}}{\mathbf{1}^T \Omega^{-1} \mathbf{1}}, \alpha = \Omega^{-1} (\mathbf{Y} - \mathbf{1}b) \quad (6)$$

在本文所定义的常矢量空间中, 输入矢量的个数 (即样本数) 可提前确定, 并不变, 而 LSSVM 的参数则需要进行优化才能确定, 一旦确定了参数, 则  $\Omega$  就是一个常矩阵。由于常矩阵  $\mathbf{A}$  和  $\mathbf{D}$  可由下式预先算出

$$\mathbf{A} = \Omega^{-1}, \mathbf{D} = \frac{\mathbf{1}^T \Omega^{-1}}{\mathbf{1}^T \Omega^{-1} \mathbf{1}} \quad (7)$$

因而式(6)可写成下式

$$b = \mathbf{D} \mathbf{Y}, \alpha = \mathbf{A} (\mathbf{Y} - \mathbf{1}b) \quad (8)$$

式中,  $\mathbf{A}$  和  $\mathbf{D}$  是事先计算出的常矩阵。由于这些矩阵只与输入矢量  $\{x_i\}$  有关, 而与输出目标  $\{y_i\}$  无关, 因此在图像处理中, 求解 LSSVM 的计算复杂度可降为  $O(N^2)$ , 且仅仅只需要进行一个矩阵相乘即可,

而且这个解是可以全局通用的。

### 2.3 图像灰度曲面最佳拟合函数

由于图像可看作曲面,因此可对原始图像像素的某邻域作曲面最佳拟合。设  $W$  和  $H$  为原始图像某一像素的对称邻域内的坐标序数集,即  $r \in W$  和  $c \in H$ ,例如  $W$  定义为  $W = \{-1, 0, 1, 2\}$ ,  $H$  定义为  $H = \{-1, 0, 1, 2\}$ ,则常矢量空间  $W \times H$  内的插值示意图如图 1 所示。设以  $(u, v)$  表示由  $\{(r, c) : -1 \leq r \leq 2, -1 \leq c \leq 2\}$  等组成的网格内任一点的位置,  $(u, v)$  为局部坐标,则对由 16 个像素组成的网格,其像素灰度的计算可在该局部坐标内进行,其 SVM 插值见图 1。在以 RBF 为核函数的 LSSVM 中,常矢量空间  $W \times H$  上的图像灰度曲面最佳拟合函数定义如下:

$$f(u, v) = \sum_{k=1}^N \alpha_k \exp \left[ -\frac{|u - r_k|^2 + |v - c_k|^2}{\sigma^2} \right] + b \quad (9)$$

式中,  $(u, v)$  为邻域中的像素坐标,  $f(u, v)$  为对应点的灰度估计值,  $(r_k, c_k)$  为观测像素坐标,  $b$  和  $\alpha_k$  为由式(8)表示的 LSSVM 解。

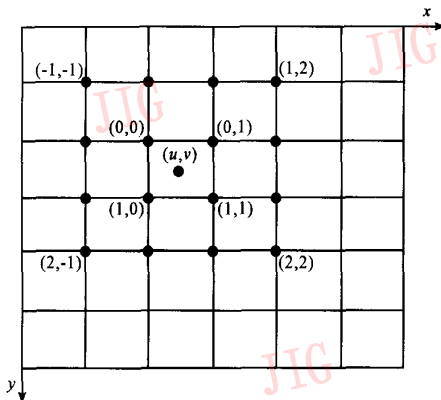


图 1 SVM 插值示意图

Fig. 1 Illustration of SVM interpolation

### 2.4 灰度插值算法

下面继续对定义在常矢量空间  $W \times H$  中的 LSSVM 解(式(8))和灰度拟合函数(式(9))进行研究。实际上,式(8)可改写为

$$\alpha = A(I - \mathbf{1D})Y \quad (10)$$

灰度拟合函数(式(9))也可进一步改写为

$$f(u, v) = F\alpha + b = FA(I - \mathbf{1D})Y + DY = [FA(I - \mathbf{1D}) + D]Y \quad (11)$$

其中

$$F = [f_1, \dots, f_N], f_k = \exp \left\{ -\frac{|u - r_k|^2 + |v - c_k|^2}{\sigma^2} \right\}, k = 1, \dots, N$$

式中,矢量  $Y$  由定义在邻域(即常矢量空间)  $W \times H$  上的图像灰度值确定。对于该邻域内的任意位置  $(u, v)$ ,  $D, A$  均由所定义的常矢量空间  $W \times H$  和核函数确定。由于它们与图像的灰度值无关,因此是可以预先计算的,而  $F$  则是与位置  $(u, v)$  有关的。式(11)就是用 SVM 算法进行图像插值的基本公式。

### 2.5 算法的快速实现——区域插值系数矩阵

由式(11)可知,当位置  $(u, v)$  不能事先确定时,则需要计算  $F$ ,可见其计算工作量还是很大的,但如果插值的位置可以预先确定时,则  $F$  是可事先计算的常矢量。这种情况下式(11)可改写为

$$f(u, v) = F\alpha + b = [FA(I - \mathbf{1D}) + D]Y = GY \quad (12)$$

式中,变形矩阵  $G$  是可以预先计算出来的。式(12)表明,对定义在常矢量空间  $W \times H$  中的已知点  $(u, v)$ ,其灰度值可以通过由该邻域像素的灰度值的线性组合来估计,而与每一个像素灰度相对应的权系数则由变形矩阵  $G$  确定。对于一个矩形邻域,变形矩阵  $G$  权系数就变成了新的灰度插值算子  $p$ ,这样不同位置的灰度插值就可独立地通过相应的算子与图像进行卷积计算而得到。如果将一个邻域内的所有插值点都按此法进行计算,以便确定与每一个插值点所对应的权系数,则可得一个由一组权系数所构成的一个矩阵  $Q$ ;然后通过系数矩阵  $Q$  与灰度矢量  $Y$  的乘积,即可得到由该区域内所有插值点的灰度估计值所构成的矢量,以完成区域内的插值运算。每一个插值点的灰度估计所需计算量为  $O(N)$ ,其中  $N$  为定义在常矢量空间  $W \times H$  中的像素点数。

### 2.6 参数的优化

由于 SVM 灰度插值算法是在基于 LSSVM 的拟合曲面上进行的,因此其与其他方法不同,即 LSSVM 中有多个参数需要进行优化确定,在本文中主要有  $\sigma^2$  和  $\gamma$  两个参数,其中 LSSVM 中的参数  $\gamma$  决定 LSSVM 解对误差的灵敏度,当它取无穷大时,则所得的 LSSVM 解即为最小二乘解,在本文讨论的问题中,为了求得最佳的拟合图像曲面,首先参数  $\gamma$  应取无穷大,而参数  $\sigma^2$  则与频率响应有关;其次邻域的大小也是很重要的。由于最优参数应该对应最佳的图像插值性能,因此要想确定参数的最优值,就必须要有个插值放大性能的评价标准,然后才能据此来优化参数。

评价插值还原性能的方法很多,其中一个方法

就是利用一已知测试图像,首先用人工进行欠采样,然后以欠采样后的图像为基础进行插值还原,同时计算峰值信噪比(peak signal noise ratio, PSNR),若峰值信噪比值越大,则算法的性能越好。

不同参数  $\sigma^2$  下的图像插值放大性能的比较是基于图像 Lena 进行的。性能比较需要针对不同参数  $\sigma^2$  来计算与不同放大倍数相对应的 PSNR,并绘制 PSNR- $\sigma^2$  曲线,最后所得的结果如图 2 所示。由图 2 可见,在邻域的大小为  $5 \times 5$  pixels 时,不同放大倍数所需的参数  $\sigma^2$  的优化值是不一样的,其中在进行 2 倍放大时,参数  $\sigma^2$  的优化值取 0.58;在进行 4 倍放大时,取 0.46;在进行 8 倍放大时,取 0.42;在进行 16 倍放大时,取 0.33 为宜。

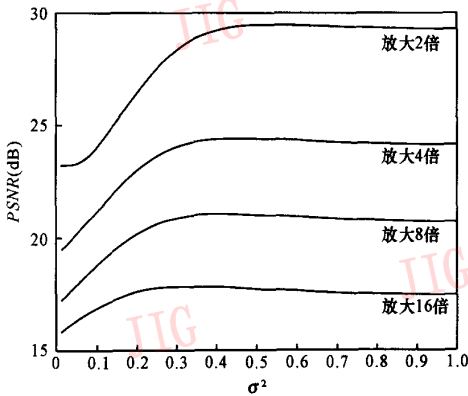


图 2 PSNR- $\sigma^2$  曲线(邻域为  $5 \times 5$  pixels)

Fig. 2 PSNR- $\sigma^2$  curve(neighborhood size is  $5 \times 5$  pixels)

### 3 实验结果和讨论

为了检验本文所提出的 SVM 图像插值算法的性能和有效性,针对灰度图像进行了大量的计算机实验。本文选了 4 幅灰度级为 256 的灰度图像(见图 3),它们的大小均为  $256 \times 256$  pixels。这些图像分别是 Lena, Pepper, Red-leaf 和 Happy,其中包括了自然和人造的图像。为了测试算法的性能,用人工对这些图像进行欠采样,首先得到了相应的缩小了 2 倍、4 倍、8 倍和 16 倍的图像;然后用本文的插值算法,取邻域的大小为  $5 \times 5$  pixels,并以上节所选取的参数值进行了放大还原,同时计算相应的峰值信噪比值,并统计所需的时间;最后在所选取的图像中选取一小块区域进行局部放大,以比较放大图像的视觉效果。所有的这些实验均在 Pentium III™ 700-MHz PC 上用 Matlab™ 6.5 完成。



图 3 大小为  $256 \times 256$  的插值放大测试图像

Fig. 3 Images used for interpolation experiments, their size are all  $256 \times 256$  pixels

#### 3.1 插值算法性能比较

几种不同插值算法放大图像的峰值信噪比及所需的时间统计比较见表 1 和表 2。

表 1 不同算法不同倍数放大图像的 PSNR 统计

Tab. 1 Comparison of PSNR for different methods

测试图像	放大倍数	PSNR (dB)			
		LSSVM	Cubic	Spline	Linear
Lena	$\times 2$	29.44	29.46	29.33	29.19
	$\times 4$	24.40	24.38	24.16	24.42
	$\times 8$	21.04	20.94	20.68	21.13
	$\times 16$	17.83	17.66	17.46	17.98
Pepper	$\times 2$	29.84	29.87	29.65	29.79
	$\times 4$	25.65	25.64	25.41	25.62
	$\times 8$	21.81	21.76	21.43	21.90
	$\times 16$	18.39	18.10	17.64	18.48
Red-leaf	$\times 2$	34.90	34.88	34.94	34.11
	$\times 4$	28.20	28.18	27.93	28.11
	$\times 8$	24.36	24.22	23.85	24.47
	$\times 16$	22.07	21.77	21.33	22.25
Happy	$\times 2$	24.51	24.49	24.31	24.28
	$\times 4$	19.25	19.27	19.05	19.16
	$\times 8$	14.55	14.61	14.67	14.60
	$\times 16$	11.66	11.34	10.87	11.58

表 1 是由不同插值算法所得到的峰值信噪比值的统计结果。由表 1 可见,当放大倍数为 2 倍时,基于 SVM 的图像灰度插值算法的性能与 Cubic 算法

表 2 不同插值算法所需的时间统计 单位:s

Tab. 2 Statistic of times for different methods

测试图像	放大倍数	算法			
		LSSVM	Cubic	Spline	Linear
Lena	×2	0.55	0.72	0.38	0.28
	×4	0.22	0.66	0.33	0.22
	×8	0.11	0.66	0.27	0.28
	×16	0.17	0.71	0.22	0.22
Pepper	×2	0.77	0.83	0.49	0.33
	×4	0.33	0.66	0.27	0.28
	×8	0.17	0.71	0.17	0.33
	×16	0.11	0.88	0.22	0.27
Redleaf	×2	0.55	0.66	0.38	0.22
	×4	0.22	0.66	0.28	0.22
	×8	0.11	0.66	0.16	0.28
	×16	0.16	0.66	0.16	0.22
Happy	×2	0.77	0.88	0.55	0.33
	×4	0.28	0.88	0.33	0.27
	×8	0.16	0.77	0.27	0.28
	×16	0.16	0.71	0.22	0.28

相当,而随着放大倍数的增大,SVM 算法放大图像的峰值信噪比略大于 Cubic 算法放大图像的峰值信噪比。以 Lena 图像为例,当放大倍数为 2 倍时,SVM 算法放大图像的峰值信噪比为 29.44dB,比 Cubic 算法放大图像的峰值信噪比 29.46dB 略小,但放大倍数为 4 倍时,SVM 算法放大图像的峰值信噪比为 24.40dB,略大于 Cubic 算法放大图像的峰值信噪比 24.38dB,放大倍数为 8 倍和 16 倍时有相似的结果。表 1 中的其他图像的放大实验结果也验证了这一结论。

为观察图像放大以后的细节,将图 2 所示的 4 幅图像中的一个尺寸为  $64 \times 64$  pixels 的局部区域分别用 SVM 算法和 Cubic 算法进行 8 倍的放大,所得的结果比较见图 4~图 6。由图 4~图 6 可见,由这两种算法进行局部放大后,其图像细节的视觉差异不明显。这进一步说明 SVM 算法与 Cubic 算法的插值放大性能相当。



(a) SVM 算法放大图像 (b) Cubic 算法放大图像

图 4 两种不同算法放大的 Lena 图像  
Fig. 4 Magnified image of Lena image



(a) SVM 算法放大图像 (b) Cubic 算法放大图像

图 5 两种不同算法放大的 Pepper 图像

Fig. 5 Magnified image of Pepper image



(a) SVM 算法放大图像 (b) Cubic 算法放大图像

图 6 两种不同算法放大的 Happy 图像

Fig. 6 Magnified image of Happy image

### 3.2 效率比较

为了比较 SVM 算法的插值效率,对不同的测试图像进行了不同倍数的插值放大实验,并统计它们所需要的时间,所得统计结果如表 2 所示。

由表 2 可见,本文中的 SVM 插值算法所需的时间小于 Cubic 算法所需要的时间,而且放大倍数越大,节省时间越明显,几乎与 Spline 算法相近。如对 Lena 图像,当放大倍数为 2 倍时,SVM 算法所需的时间为 0.55s,明显小于 Cubic 算法所需的时间 0.72s;而当放大倍数为 4 倍时,SVM 算法所需的时间为 0.22s,大约是 Cubic 算法所需时间 0.66s 的三分之一。其他图像的实验结果也证实了这一点。这些实验结果表明,SVM 插值算法有比 Cubic 算法更高的效率。

## 4 结 论

本文首先利用最小二乘向量机对图像中每个局部区域进行了图像灰度曲面的最佳函数拟合,然后在图像灰度拟合曲面上进行灰度的插值估计,同时运用映射技术将在图像区域上的 LSSVM 学习问题转换至一个常矢量空间来实现,这样完成 LSSVM 的训练仅仅只需要一个常矩阵相乘即可,且这个一次性的学习结果可以推广到整幅图像。当放大倍数

确定后,即可将一个区域的插值运算转换成一个插值系数矩阵与该区域中的灰度矢量的乘法运算。文中以 RBF 核函数为例导出了图像的区域插值系数矩阵,并提出了由测试图像进行参数优化的方法。最后用优化的参数对图像进行了插值放大实验。实验结果表明,本文所提出的 SVM 图像插值算法不但与 Cubic 算法具有相近的性能,且具有更高的效率。

#### 参考文献 (References)

- 1 Maeland E. On the comparison of interpolation methods[J]. IEEE Transactions on Medical Imaging, 1988,7(3): 213 ~ 217.
- 2 Keys R. Cubic convolution interpolation for digital image processing [J]. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1981,29(6): 1153 ~ 1160.
- 3 Yao Haigeng. Image processing[M]. Shanghai, China: Shanghai Scientific and Technical Publishers, 2001, 96 ~ 106. [姚海根. 图像处理[M]. 上海:上海科学技术出版社,2001, 96 ~ 106.]
- 4 Nene S A, Nayar S K. A simple algorithm for nearest neighbor search in high dimensions[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997,19(9): 989 ~ 1003.
- 5 Vapnik V. The nature of statistical learning theory[M]. New York, USA: Wiley, 1998, Chapter 5.
- 6 Suykens J A K, Vandewalle J. Least squares support vector machine classifiers[J]. Neural Processing Letters, 1999,9(3): 293 ~ 300.