

考虑非理想排错过程的软件可靠性模型

郑 垒, 沈元隆

(南京邮电大学 电子科学与工程学院, 江苏 南京 210003)

摘 要:经典的可靠性模型排错过程是理想的,没有考虑各种实际情况。实际的排错过程并不是完美的,错误排除需要时间,且不可能完全排除,排错过程中可能引入新的错误,错误排除率和错误引入率均是随时间变化的函数等等。文中针对这些排错过程的实际情况,对 Xuemei Zhang 等人提出的软件可靠性模型进行了改进,提出新的假设,建立改进的新模型,给出模型的一般表达形式。并通过两组公开发表的失效数据,对改进后模型的一个特例模型的拟合预测能力进行仿真分析比较,最终验证了改进的考虑非理想排错过程的软件可靠性模型的优越性,说明了新模型更符合现实的软件可靠性活动过程。

关键词:软件可靠性模型;排错过程;错误排除率函数;错误引入率函数;排错时间

中图分类号:TP311.56

文献标识码:A

文章编号:1673-629X(2011)08-0112-05

A Model of Software-Reliability with Imperfect Fault Correction Process

ZHENG Lei, SHEN Yuan-long

(Institute of Electronics Science and Engineering, Nanjing University of Posts and Telecommunications,
Nanjing 210003, China)

Abstract: The classical models, which were based on ideal fault correction process, ignore the effect of the realistic fault correction process. In fact, the realistic fault correction process is not perfect. The fault can not be corrected immediately after it be detected, and the fault introduction and removal efficiency should be considered. The fault removal rate is not constant, but depends on time. In this paper, considering the actual impact of fault correction process, some new assumptions are proposed, and a new improved model based on the software reliability model which was proposed by Xuemei Zhang is created. Some simulation experiments are carried out in the new model and selected ones with two set of software failures data which were openly published. Through analyzing and comparing one of the improved model, it ultimately proves the superiority of the improved prediction model, and the new models is more suitable to be used in actual software reliability process.

Key words: software reliability model; fault correction process; fault correction rate function; fault introduction rate function; fault correction time

0 引言

软件可靠性模型是评估软件可靠性的主要工具。模型利用软件测试时软件失效的历史信息,预测软件实际工作时的现场行为,从而决定该软件可靠性以及是否可以投放使用。

错误发现过程和错误排除过程是影响软件可靠性建模准确性的两个主要因素^[1]。现有的模型主要集中于对错误发现过程的精确建模,对错误排除过程的研究并不充分^[2]。对软件排错过程更精确的描述建模可

以显著提高软件可靠性模型的评价能力和预测能力,而准确地评价和预测软件可靠度有助于确定软件的合理释放时间,节省软件测试费用和开发测试时间,对软件工程管理有指导作用^[3]。

非齐次泊松过程(NHPP)类模型^[4]是软件可靠性增长模型中最常见也是最重要的一类模型,G-O模型^[5]作为非齐次泊松过程(NHPP)类模型的经典模型应用最为广泛。这类模型一般假设错误一旦发现即刻完全排除,并且排除错误时不会引入新的错误,而且排错时间忽略不计。这些假设条件都过于理想化,使可靠性模型变得简单,但同时也降低了模型精度和结果的可信度。

文中在 Xuemei Zhang, Xiaolin Teng 和 Hoang Pham 提出的软件可靠性模型^[6](以下简称 ZTP 模型)

收稿日期:2011-01-12;修回日期:2011-04-23

基金项目:南京邮电大学科研基金(01 院 32)

作者简介:郑 垒(1988-),男,硕士研究生,主要研究方向为软件可靠性;沈元隆:教授,硕士生导师,主要研究方向为通信系统的可靠性技术。

的基础上,考虑非理想排错过程的影响,提出新的假设,建立改进的新模型,通过两组公开发表的失效数据进行曲线拟合比较。结果表明,文中提出的模型能够取得更好的拟合和预测效果。

1 模型假设

ZTP模型很好地综合了软件的检查 and 排错过程,适用广泛,具有良好的精确度和预测能力。但ZTP模型中也存在不足:虽然模型中考虑了错误排除率和错误引入率,但其皆为常数,这与实际的软件开发过程不符。因此,考虑实际中非理想排错过程,可以从以下三个方面对ZTP模型加以综合改进:

1) 错误排除需要一定的时间,忽略排错时间的假设过于苛刻,尤其在软件工程的实践中^[7]。在实际的测试中,不同错误的排除时间差别很大,错误的严重程度不同,排除时间也不同。因此,建立软件可靠性模型需要考虑排错时间的影响。

2) 错误排除率不是100%,也不是常数,根据实际,随着测试的进行,软件中隐藏的错误的类别结构也会发生变化,因此错误排除效率可以看作测试时间的函数。

3) 排错过程中会引入新的错误,为了更符合实际,错误引入率同错误排除率一样,不是常量,它会随着测试的进行发生变化,因此也可以看作测试时间的函数。

基于以上的讨论,在ZTP模型的基础上,可以建立考虑非理想排错过程的可靠性模型的基本假设为:

(1) 软件是在与预期操作环境相似的条件下行;

(2) 差错的查出过程是一个非齐次泊松过程;

(3) 软件失效是由残存在软件中的差错引起的;

(4) 发现错误的概率与软件中剩余错误数成正比,这一比例(错误检出率)是随时间变化的函数,以 $b(t)$ 表示, $0 < b(t) \leq 1$;

(5) 每次软件失效是由一个错误引起的,当失效发生后,错误排除需要时间,以 $\varphi(t)$ 表示, $\varphi(t) > 0$ (新提出的假设);

(6) 测试阶段中发现故障后,排错过程未必能真正排除错误,并且排除错误的概率是随时间变化的函数,以 $p(t)$ 表示, $0 < p(t) \leq 1$ (改进的假设);

(7) 排错时可能引入新的错误,且引入新错误的概率与检测到的错误数成正比,这一比例(错误引入率)是随时间变化的函数,以 $q(t)$ 表示, $q(t) \geq 0$ (改进的假设);

(8) 累积错误数的期望函数 $m(t)$ 是一个有界的单调增函数,并满足 $m(0) = 0$,且 $\lim_{t \rightarrow \infty} m(t)$ 为有限值。

2 模型建立

记 $m(t)$ 表示在时刻 t 发现的累计故障数, a 表示软件中潜伏的初始故障数, $x(t)$ 表示在时刻 t 软件累计排除的净故障数(即累计排除的故障数减去累计引入的故障数)。根据假设和原始的G-O模型,可以得到如下方程:

$$\frac{dm(t)}{dt} = b(t)[a - x(t)] \quad (1)$$

$$\frac{dx(t)}{dt} = [p(t) - q(t)] \frac{dm(t)}{dt} \quad (2)$$

初始条件为

$$m(0) = 0 \quad (3)$$

$$x(0) = 0 \quad (4)$$

将式(2)代入式(3),可得

$$\frac{dx(t)}{dt} = b(t)[p(t) - q(t)][a - x(t)] \quad (5)$$

式(5)可化为:

$$\begin{aligned} \frac{dx(t)}{dt} + b(t)[p(t) - q(t)]x(t) \\ = ab(t)[p(t) - q(t)] \end{aligned} \quad (6)$$

结合式(4),解此一阶微分方程可得

$$x(t) = e^{-\int_0^t b(\tau)(p(\tau)-q(\tau))d\tau} \int_0^t ab(\tau)e^{\int_0^\tau b(\nu)(p(\nu)-q(\nu))d\nu} d\tau \quad (7)$$

将式(7)代入式(1),结合初值式(3)可得

$$m(t) = \int_0^t b(\tau) \left\{ a - e^{-\int_0^\tau b(\nu)(p(\nu)-q(\nu))d\nu} \left[\int_0^\tau ab(\theta)e^{\int_0^\theta b(\nu)(p(\nu)-q(\nu))d\nu} d\theta \right] \right\} d\tau \quad (8)$$

用 $\lambda(t)$ 表示失效率函数,则

$$\lambda(t) = m'(t) = b(t) \left[a - e^{-\int_0^t b(\nu)(p(\nu)-q(\nu))d\nu} \int_0^t ab(\theta)e^{\int_0^\theta b(\nu)(p(\nu)-q(\nu))d\nu} d\theta \right] \quad (9)$$

软件发生第 n 次失效后可靠性函数为

$$R_{n+1}(x|T_n=s) = \Pr\{X_{n+1} > x|T_n=s\} = \exp\{-[m(x+s) - m(s)]\} \quad (10)$$

3 模型中相关函数的选取

模型中相关函数的不同,直接决定模型的类型不同,从而导致模型的精确度与预测能力的差异。软件可靠性模型建立的基本原则就是要符合实际。因此,提出的新模型中相关函数的选择的标准也是尽可能的贴近实际,符合现实的软件开发过程。下面将对新提出的模型中的相关函数进行分析讨论,选择最佳的拟合函数。

3.1 错误发现率函数

实际上,测试阶段软件的错误发现率变化应该是

故障本身的性质和测试人员的学习过程两方面作用的结果,综合这两方面的影响,可得,错误发现率的变化类似于一个 S 形的增函数^[8],其形式为

$$b(t) = \frac{b^2 t}{1 + bt}, \quad 0 < b \leq 1 \quad (11)$$

3.2 错误排除时间

错误排除时间的长短由错误性质决定。负指数分布能够较好地描述许多人工活动的服务时间;文献[9]对 178 个排错时间样本进行了分析,认为排错时间服从负指数分布,同样,通过 Shuttle 项目中包含 85 个数据项的排错时间样本分析,其排错时间也服从参数为平均排错时间的负指数分布。由于负指数分布的排错时间具有较为广泛的代表性,因此,假设排错时间服从参数为 μ_0 的负指数分布。然而,由于每个错误的排错时间不同,需要将所有的排错时间代入计算,使分析求解模型非常复杂。为了简化起见,假设所有的排错时间均为排错时间的均值 μ , $\mu \geq 0$, 即 $\varphi(t) = \mu$ 。在考虑了错误排除时间后,其纠正的累计错误数要相应地向右平移 μ 个单位。

3.3 错误排除率函数

错误排除率函数同错误发现率函数有很大的相似性,但也有区别。随着测试的进行软件中隐藏的错误的类别结构会发生变化,软件的排错效率也会发生相应的变化。经过较长时间的测试之后,随着简单容易的错误先被排除,软件中隐藏的容易排除的错误所占的比例会逐渐降低,较难或无法排除的错误会逐渐累积,因此软件的错误排除效率会逐渐降低。随着测试的进行,越晚被发现的软件错误,被定位和修改的难度越大,当测试时间足够长时,错误排除效率趋近于 0。因此软件排错效率是测试时间的函数,而且呈递减的趋势^[10]。因此根据实际,此函数应满足以下条件:

(1) 当 $t = 0$ 时, $p(t) = p$, $0 < p \leq 1$, p 为测试开始时的错误排除率;

(2) 当 $t \rightarrow \infty$ 时, $p(t) \rightarrow 0$;

(3) $p(t)$ 是减函数。

根据上述条件,可以选择一些简单有效的函数来拟合定义排错效率 $p(t)$ 。首先想到反比例函数符合上述条件,可得

$$p(t) = \frac{p}{1 + kt}, \quad k \geq 0 \quad (12)$$

其中, k 决定排错率的变化快慢。因为错误排除效率在短时间内不可能有明显的下降,反比例函数的变化过于迅速,负指数型的减函数相比反比例函数的变化趋势较为缓慢,能够较好符合实际的排错过程。因此可以选择负指数型的减函数来定义错误排错效率:

$$p(t) = pe^{-vt}, \quad v \geq 0 \quad (13)$$

其中, v 决定排错率的变化快慢。虽然排错人员也可以在排错过程中通过学习提高错误排除率,但是,实际中的排错人员一般是软件的研发设计人员,对所开发的软件非常了解,而且经过了开发过程的学习之后,在测试阶段其排错能力基本保持不变。

考虑错误排除时间的影响,从检测出一个错误到排除这个错误的 μ 时间内,此错误重复出现的次数为 $\mu \cdot b(t)$, 因此错误的排除率变为^[11]

$$p(t) = \frac{p_0(t)}{1 + ub(t)} \quad (14)$$

其中 $p_0(t)$ 为上面讨论给出的错误排除效率函数。

3.4 错误引入率函数

因为调试过程本身固有的不完美的特点,所以排除错误时,会引入一定的错误。实际中的错误引入率不可忽略,很显然,在测试的不同时刻和阶段,错误的引入率不同。实际中,错误的引入数与排错的数近似呈正比,因此,错误引入率与错误排除率有类似的变化趋势,可近似表示为

$$q(t) = \frac{q}{1 + kt}, \quad k \geq 0 \text{ 或 } q(t) = qe^{-vt}, \quad v \geq 0 \quad (15)$$

同时,也要考虑排错时间的影响,错误引入率变为

$$q(t) = \frac{q_0(t)}{1 + ub(t)} \quad (16)$$

其中 $q_0(t)$ 为上面讨论给出的错误排除效率函数。

4 模型的特例

将上节分析的得到的函数: $b(t) = \frac{b^2 t}{1 + bt}$, $p(t) = \frac{pe^{-vt}}{1 + ub(t)}$, $q(t) = \frac{qe^{-vt}}{1 + ub(t)}$ 代入式(8)中 $m(t)$ 的表达式,可得到测试过程中发现的累积错误数 $m(t)$ 的具体形式,然后可求得失效率 $\lambda(t)$ 和可靠度 $R_{n+1}(x | T_n = s)$ 的具体表达形式。

由于模型综合考虑了实际的测试排错过程,需要用到的参数很多,模型因此变得复杂,给方程的求解和参数估计带来困难。新模型中,由于选择的相关函数不同,可能出现检测到的累计错误数无法用初等函数表示的情况,利用数值分析知识可以求近似解,此时的参数估计可以通过先估计出起始初值,然后用迭代的优化算法如遗传算法或粒子群算法进行求解。

为了便于求解和参数估计,对此新提出的模型进行简化,考虑模型的一种特例情况:

(1) 排错率为常数,即

$$b(t) = b, \quad 0 < b \leq 1 \quad (17)$$

(2) $\varphi(t) = \mu$ (18)

(3)
$$p(t) = \frac{p}{1 + kt} = \frac{p}{(1 + kt)(1 + \mu b)}$$
 (19)

(4)
$$q(t) = \frac{q}{1 + kb(t)} = \frac{q}{(1 + kt)(1 + \mu b)}$$
 (20)

将式(17)、(18)、(19)和(20)代入式(7)可得

$$x(t) = a[1 - (kt + 1)^{-\frac{p+q}{p+q+\mu b}}]$$
 (21)

将式(21)代入式(8)可得累计错误数为

$$m(t) = \frac{ab(1 + \mu b)[(kt + 1)^{1 - \frac{p+q}{p+q+\mu b}} - 1]}{k(1 + \mu b) - b(p + q)}$$
 (22)

在应用上述结论时,需对 a, b, p, q, k, μ 做出估计,可以用最大似然或最小二乘法进行参数估计。

5 模型验证与评估

5.1 仿真使用的模型

文中将以 G-O 模型和 ZTP 模型为基础,对比分析评估提出的新模型的性能。ZTP 模型的累积错误数表达式为

$$m(t) = \frac{a}{p - \beta} [1 - (\frac{1 + \alpha}{1 + \alpha e^{-bt}})^{\frac{a}{p - \beta}}]$$
 (23)

其中, α, b 和 c 为体现学习过程的故障检测率函数中的三个参数, p 为错误发现率, β 为错误排除率。

下面将通过两组实际数据对文中提出的模型进行验证评估,所用的数据均来自实际工作项目中的数据。需要注意的是,为了便于计算,验证所选用的模型是新模型的一种特例,在有足够计算能力的条件下,计算新模型得到的验证结果应该比所选用的特例模型情况要更好。

5.2 仿真验证分析(1)

本组仿真试验的失效数据集来自文献[12]提供的 Tandem 计算机公司。对各模型经过参数估计,得到原始数据、模型预测及仿真结果如图 1 所示。

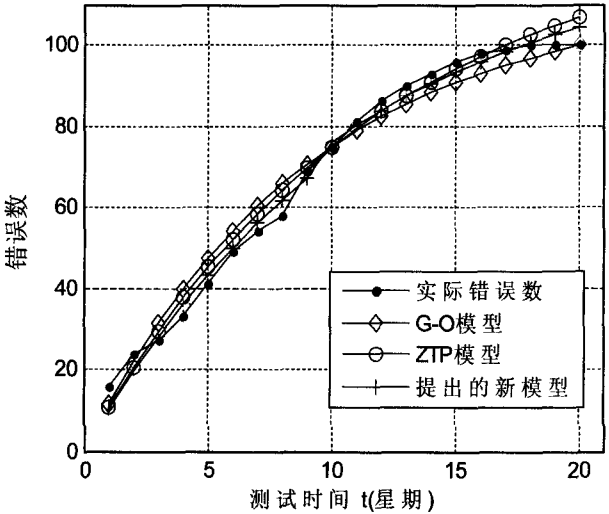


图 1 G-O 模型,ZTP 模型和新模型对实际数据的拟合

从图 1 中可以看出,从拟合数据能力上看,新提出的模型最好,ZTP 模型其次,G-O 模型最差,改进后的模型得到的错误数比原来的 ZTP 模型更接近实际数据。说明新提出的模型与 G-O 模型和 ZTP 模型相比,能够更精确地对失效数据进行拟合。

模型的评价标准为:

- (1) 误差平方和 SSE 越小,则表示曲线的拟合能力越好;
- (2) 回归曲线方程的相关指数 R-Square 越接近于 1,则表示曲线的拟合能力越好;
- (3) 平均误差 AE 越小,表示模型的预测能力越强。

因此,从表 1 可以看出,提出的模型的这三个比较标准值都优于其他两个模型,说明该模型对失效数据具有良好的拟合性和预测能力。

表 1 G-O 模型、ZTP 模型和新模型比较结果

	SSE	R-Square	AE
G-O 模型	403.3044	0.8479	0.0813
ZTP 模型	257.6096	1.0280	0.0681
提出的新模型	154.4617	1.0221	0.0570

5.3 仿真验证分析(2)

本组仿真试验的失效数据集来自文献[13]中实际测得的软件失效数据。对各模型经过参数估计,得到原始数据、模型预测及仿真结果如图 2 所示。

同样,从图 2 中可以看出,文中提出的新模型的累积错误数曲线与实际的累积错误数曲线拟合的最好,ZTP 模型其次,G-O 模型最差。

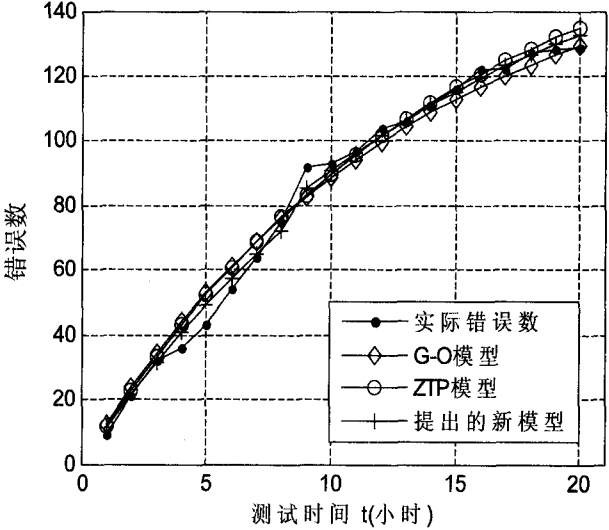


图 2 G-O 模型,ZTP 模型和新模型对实际数据的拟合

从表 2 可以看出,提出的新模型的 SSE, R-Square 和 AE 值都优于其他两个模型,。因此,该模型对失效数据具有良好的拟合性和预测能力,体现了文中提出的改进的新模型的优越性。

表 2 G-O 模型、ZTP 模型和新模型比较结果

	SSE	R-Square	AE
G-O	470.1561	0.8367	0.0827
ZTP 模型	380.4837	0.9454	0.0698
提出的新模型	174.0872	0.9545	0.0452

6 结束语

迄今为止,还没有一个通用的软件可靠性模型可以适用于任何一个软件开发的过程当中。因此只能根据实际情况,选择适当的可靠性模型对软件开发过程进行预测。文中所提出的模型,综合考虑了排错时间、排错效率和错误引入率不是理想情形的影响,此该模型更为准确地体现了非理想排错过程的特点,使模型更加符合实际情况。通过两组公开发表的软件失效数据对模型进行了拟合分析,验证了新模型的对软件可靠性的评估和预测能力,而且所验证的模型仅是新提出的一般模型的简化特例。对于文中提出使用的错误发现率函数,错误排除效率函数,错误引入率函数等函数的具体形式,还可以根据其性质和实际的具体软件开发过程选取其他的函数来建立求解模型。

参考文献:

- [1] Lo Jung-Hua. Considering Both Failure Detection and Fault Correction Activities in Software Reliability Modeling [C]//IEEE Region 10 International Conference (TENCON 2006). [s. l.]: [s. n.], 2006:1-4.
- [2] Lo J, Huang C. An integration of fault detection and correction processes in software reliability analysis [J]. Journal of System and Software, 2006(79):1312-1323.
- [3] 谢景燕, 安金霞, 朱纪洪. 考虑不完美排错情况的 NHPP 类软件可靠性增长模型 [J]. 软件学报, 2010, 21(5): 942-949.
- [4] Huang C Y, Lyu M R, Kuo S Y. A unified scheme of some nonhomogenous Poisson process models for software reliability estimation [J]. IEEE Transaction on Software Engineering, 2003, 29(3): 261-269.
- [5] Goel A L, Okumoto K. Time dependent error detection rate model for software reliability and other performance measures [J]. IEEE Transaction on Reliability, 1979, 28(3): 206-211.
- [6] Zhang Xuemei, Teng Xiaolin, Pham H. Considering Fault Removal Efficiency in Software Reliability Assessment [J]. IEEE Transactions on System, Man, and Cybernetics, 2003, 33(1): 114-119.
- [7] Huang C, Lin C. Software reliability analysis by considering fault dependency and debugging time lag [J]. IEEE Transaction on Reliability, 2006, 55(3): 436-450.
- [8] 吴彩华, 朱小冬, 刘俊涛. 一种新的软件可靠性增长模型 [J]. 系统工程与电子技术, 2009, 31(8): 2024-2028.
- [9] Musa J D. Software reliability measurement, prediction, application [M]. New York: McGraw-Hill Press, 1987.
- [10] Huang C Y, Kuo S Y, Michael R L, et al. Quantitative Software Reliability Modeling from Testing to Operation [C]//11th International Symposium on Software Reliability Engineering. [s. l.]: [s. n.], 2000: 72-82.
- [11] 刘宏伟, 杨孝宗, 岳晓光. 一个考虑多种排错延迟的 NHPP 类软件可靠性增长模型 [J]. 计算机工程与应用, 2004(30): 12-14.
- [12] Wood A. Predicting software reliability [J]. IEEE Computer, 1996, 29(11): 69-77.
- [13] Pham H, Zhang X. A Software Cost Model with Warranty and Risk Costs [J]. IEEE Trans. Computers, 1999, 48(1): 71-75.
- [4] Wang Jianan, Kong Jun. A region-based SRG algorithm for color image segmentation [C]//Proceedings of the Sixth International Conference on Machine Learning and Cybernetics. Hong Kong: [s. n.], 2007: 19-22.
- [5] 林开颜, 徐立鸿, 吴军辉. 快速模糊 C 均值聚类彩色图像分割方法 [J]. 中国图像图形学报, 2004, 9(2): 159-163.
- [6] 周礼平, 高新波. 图像分割的快速模糊 C 均值聚类算法 [J]. 计算机工程与应用, 2004(8): 68-70.
- [7] Zhang Yongyue, Brady M, Smith S. Segmentation of Brain Image Through a Hidden Markov Random Field Model and the Expectation-Maximization Algorithm [J]. IEEE Transaction on Medical Image, 2001, 20(1): 45-57.
- [8] 刘盈盈, 石跃祥, 莫浩瀚. 基于改进 K 均值算法在彩色图像分割中的应用 [J]. 计算机工程与应用, 2008, 44(29): 191-192.
- [9] 王志豪, 汪继文. 一种基于 CB 模型的彩色图像分割方法 [J]. 计算机技术与发展, 2008, 18(5): 44-46.
- [10] Gao Pengdong, Lu Yongquan, Qiu Chu. Performance Comparison between Color and Spatial Segmentation for Image Retrieval and Its Parallel System Implementation [J]. Journal of Liaoning Normal University, 2008, 30(3): 539-543.
- [11] Fernandes S D F, Monteiro A M V. Parallelism and images: a parallelization experiment for image segmentation with an application for automatic classification of scenes obtained from orbital platforms [C]//Proceedings XIII Brazilian Symposium on Computer Graphics and Image Processing. [s. l.]: [s. n.], 2000: 342-344.
- [12] Kenney C, Deng Y, Manjunath B S. Peer group image enhancement [J]. IEEE Transaction on Image Processing, 2001, 10(2): 326-334.

(上接第 111 页)