

基于预检验的快速随机抽样一致性算法*

陈付幸⁺, 王润生

(国防科学技术大学 ATR 国家重点实验室, 湖南 长沙 410073)

Fast RANSAC with Preview Model Parameters Evaluation

CHEN Fu-Xing⁺, WANG Run-Sheng

(National Key Laboratory of ATR, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: Phn: +86-731-4573468, E-mail: cft911@163.com, <http://www.nudt.edu.cn>

Received 2004-01-09; Accepted 2005-02-04

Chen FX, Wang RS. Fast RANSAC with preview model parameters evaluation. *Journal of Software*, 2005,16(8):1431-1437. DOI: 10.1360/jos161431

Abstract: RANSAC algorithm is one of the most widely used robust estimator in the field of computer vision, but, it's efficiency is low. The paper gives a preview model parameters evaluation RANSAC algorithm (PERANSAC): a preview model parameters evaluation selection is added to the RANSAC algorithm. With guaranteeing the same confidence of the solution as RANSAC, a very large number of erroneous model parameters obtained from the contaminated samples are discarded in the preview evaluation selection. PERANSAC algorithm is evaluated on both synthetic data and real-world images, a significant increase in speed is shown, and the solutions are the same as RANSAC's.

Key words: RANSAC; PERANSAC; robust; fundamental matrix; LmedS estimation; outliers; inliers

摘要: 随机抽样一致性算法 RANSAC(random sample consensus)是在计算机视觉领域内应用最广泛的 Robust 估计算法之一,但是 RANSAC 算法计算效率较低.提出一种基于模型参数预检验的快速 RANSAC 算法(preview model parameters evaluation RANSAC,简称 PERANSAC).算法在 RANSAC 算法模型参数检验前,加入预检验过程,在保证计算结果置信概率不变的前提下,过滤掉大量偏差较大的模型参数,提高了 RANSAC 算法的整体效率.模拟数据和真实图像数据实验结果表明,此算法和 RANSAC 算法计算精度保持一致,计算速度高于 RANSAC 算法.

关键词: RANSAC;PERANSAC;鲁棒性;基础矩阵;LmedS 估计;outliers;inliers

中图法分类号: TP391 文献标识码: A

在计算机视觉以及其他很多研究领域,模型参数的 Robust 估计是一个核心问题,问题的关键在于对不符合实际模型的 outliers 处理上,统计学家认为 outliers 是整体上与假设模型不符合的数据,inliers 是假设模型符合的数据,通常,outliers 分为大误差数据和完全错误数据.虽然人们在这方面作了大量的研究,但是至今还没有一个 Robust 估计方法,可以做到完全消除 outliers 对估计的影响^[1].

M 估计法^[2,3]用一个余差的函数来代替最小二乘算法中的余差平方,以此抑制大余差对估计过程的影响.M

* 作者简介: 陈付幸(1976 -),男,河南新野人,博士,主要研究领域为图像分析与理解,基于图像的三维建模;王润生(1941 -),男,教授,博士生导师,主要研究领域为图像分析与理解,模式识别,信息融合.

估计算法一般通过加权迭代求解,对初值依赖比较大,初值一般由最小二乘法获得,受错误数据影响较大.所以 M 估计算法对大误差数据有较好的抑制作用,但是对完全错误数据,此算法不适用.最小中值 Least-Median-Squares(LMedS)^[2,3]算法通过最小化余差平方中值来估计模型参数,实验证明,LMedS 算法对 outliers 的鲁棒性比较好.由于 LMedS 算法是通过最小化余差平方中值求解,所以,当数据错误率超过 50%时,此算法不再适用.

由 Fishler 和 Bolles^[4]提出的 RANSAC 算法,对错误率超过 50%的数据仍然能够处理,是最有效的 Robust 估计算法之一,在计算机视觉领域得到了广泛的应用.例如,基础矩阵估计^[1-3,5]、特征匹配^[6]、运动模型选择^[7]等.同时,根据随机抽样的思想,衍生了很多 Robust 算法,例如 MINPRAN^[8],MLEAC^[9]等.

在 RANSAC 算法的模型参数检验中, M 组抽样大部分都受到 outliers 影响,全部参与全数据检验必然会造成计算浪费.如果能减少参与全数据检验的模型参数数量,就会提高 RANSAC 算法的效率.本文提出了 PERANSAC(preview model parameters evaluation RANSAC)算法:在对每个模型参数检验之前,首先随机抽取少量的数据对模型参数进行预检验,如果模型参数通过预检验,则进入后续的检验,否则这个模型参数不再参与后续的全数据检验.同时,算法通过增加抽样数量 M ,保证算法结果具有和 RANSAC 算法一样的置信度.虽然增加了抽样数量 M ,但是,由于大部分模型参数都受到错误数据的影响,经过预检验,过滤掉的模型参数数量远远超过增加的模型参数,减少了进行全数据检验的模型参数数量,提高了计算效率.

1 RANSAC 算法

1.1 RANSAC算法的计算量

RANSAC 算法中,要求保证在一定的置信概率下, M 组抽样中至少有一组抽样的数据全是 inliers.利用式(1),可以求得满足要求的最小抽样数 M .

$$1 - (1 - (1 - \varepsilon)^m)^M = P \quad (1)$$

其中, ε 为数据错误率(outliers 在原始数据所占的比例), m 为计算模型参数需要的最小数据量, P 为置信概率.从式(1)可以看出, M 和 ε , m , P 呈指数关系.表 1 所示为 $P=0.95$ 时, M 随 ε , m 变化的情况.

Table 1 The number M of samples required to ensure $P=0.95$ for given ε and m

表 1 置信概率 $P=0.95$ 时, M 随 ε, m 变化的情况

| Dimensionality | Contamination levels | | | | | | |
|----------------|----------------------|-----|-----|-----|-----|-----|------|
| | 5% | 10% | 20% | 25% | 30% | 40% | 50% |
| 4 | 2 | 3 | 6 | 8 | 11 | 22 | 47 |
| 6 | 3 | 4 | 10 | 16 | 24 | 63 | 191 |
| 8 | 3 | 6 | 17 | 29 | 51 | 177 | 766 |
| 9 | 4 | 7 | 21 | 39 | 73 | 296 | 1533 |
| 10 | 4 | 7 | 27 | 52 | 105 | 494 | 3067 |

从表 1 可以看出,当模型比较复杂、 ε 较高时, M 很大,直接造成 RANSAC 算法效率下降.

设从原始数据中随机抽取一组抽样需要时间 t_s ;由一组抽样计算模型参数需要时间 t_c ;用一个数据检验模型参数需时间平均为 t ,则用 N 个数据检验(全数据检验)需要时间 Nt .因此 RANSAC 算法所需计算时间为

$$T = M(t_s + t_c) + MNt \quad (2)$$

其中 $M(t_s + t_c)$ 为 M 组抽样抽取及模型参数计算需要的时间, MNt 为 M 个模型参数检验需要的时间.

从式(2)看出,RANSAC 算法需要的时间由两部分组成:(1) M 组抽样选择和模型参数估计需要的时间;(2) 模型参数检验需要的时间.在模型确定、数据错误率确定的情况下,为了保证结果的置信概率, M 是不能减少的.因此,为了提高算法的效率,只能从减少参与检验的模型参数数量、减少模型参数检验需要的时间出发.

1.2 RANSAC算法步骤

- (1) 由式(1),根据置信概率 P 和数据错误率 ε 计算最小抽样数 M ;
- (2) 计算抽样对应的模型参数,用所有原始数据检验模型参数质量,获得每个模型参数的 inliers 数量;
- (3) 根据 inliers 数量和误差的方差来选择最优模型参数;

(4) 用最优模型参数对应的 inliers 估计最终模型参数.

2 PERANSAC 算法

2.1 算法核心思想

为了减少参与检验的模型参数数量,在全数据检验之前,用少量随机抽取的数据对模型进行预检验,能够通过预检验的模型参数才能参与全数据检验.

- 用 inliers 数量作为检验模型参数质量的标准和检验所用样本数无关,因此在小样本数据检验下,此标准仍然成立.

- 大部分模型参数都受到 outliers 影响,预检验过程只需从原始数据中随机选择少量数据就可以检验出不可能是最后结果的模型参数,过滤掉大量错误模型参数.

- 增加抽样数 M , 保证至少有一个正确模型参数(计算该模型参数的数据全为 inliers)能够通过预检验.

由式(1)得知,一个模型参数是正确模型参数的概率为 $(1-\varepsilon)^m$, 经过预检验后,正确模型参数存在的一定概率被误认为是错误的模型参数,被过滤掉,设 P_f 为正确模型参数通过预检验的概率,则通过预检验后,一个模型参数为正确模型参数的概率为 $(1-\varepsilon)^m P_f$, 则式(1)变为

$$1 - (1 - (1 - \varepsilon)^m P_f)^M = P \quad (3)$$

用式(3)完全可以描述,只要抽取 M 组抽样,就可以在置信概率 P 下,保证通过预检验后,保留的模型参数中至少有一个正确模型参数.这保证了经过预处理后,算法的置信概率和 RANSAC 算法的置信概率是一样的,保证了算法的准确性.当没有预检验过程时, $P_f=1$, 对应式(1),就是 RANSAC 算法.

这样,通过式(3)把预检验过程对计算结果的影响转移到抽样抽取过程中去,当 $P_f < 1$ 时,对应需要抽取的抽样数量 M 比 RANSAC 算法要多一些,但由于大部分模型参数受到 outliers 影响,相对预检验过程可以过滤掉的模型参数来说,增加的模型参数是很少的.所以,加入预检验过程后,在不改变 RANSAC 算法计算精度的情况下,减少了计算量,提高了算法效率.

2.2 预检验过程

假设进行预检验的样本数据为 n , 至少存在 n_f 个 inliers 的概率为 P_f , 这 n_f 个数据对正确模型参数是一致的.也就是说,对正确模型参数,在置信概率 P_f 下,用 n 个数据进行检验时,至少有 n_f 个数据是 inliers, 如果 inliers 数量少于 n_f , 说明这个模型参数不是正确模型参数,受到 outliers 影响,则这个模型参数不再参与比较.对正确模型参数来说,通过预检验的概率 P_f 为

$$1 - \sum_{i=0}^{n_f-1} C_n^i \varepsilon^{(n-i)} (1-\varepsilon)^i = P_f \quad (4)$$

式中 C_n^i 为从 n 个数据中选择 i 个数据的组合.在实际计算中,设定 P_f 的最低限,通过试探不同的 n_f , 计算得到满足条件的最大 n_f 和对应的 P_f .

根据误差项标准偏差来检验数据是否是模型参数的 inliers^[8,9], 如式(5), 当数据对应的误差小于误差项标准偏差的 1.96 倍时,认为该数据是 inliers, 否则为 outliers^[8,9].

$$z = \begin{cases} \text{inliers, } |d| \leq t = 1.96\sigma \\ \text{outlier, otherwise} \end{cases} \quad (5)$$

当 $n \geq 2m$ 时,误差项标准偏差 σ 可估计如下:

$$\sigma = 1.4826 \left(1 + \frac{5}{n-m} \right) \sqrt{\text{med}_i |d_i|} \quad (6)$$

式中 d_i 为数据对应的误差项, $\text{med}_i |d_i|$ 为取误差项绝对值的中值, n 为检验的样本数, m 为计算模型参数需要的小数据量.

对每个模型参数,由式(5)得到模型参数对应的 inliers 数量,然后与 n_f 比较,如果不小于 n_f , 则该模型参数通过

预检测,否则该模型参数不能通过预检测.

2.3 PERANSAC算法减少的计算量 ΔT

设错误模型参数(计算模型参数的数据中包含有 outliers)通过预检验的概率为 P_{error} , 一个模型参数预检验需要的时间为 t' , 则 PERANSAC 算法需要的计算时间为

$$T' = (M + \Delta M)(P_f P_{mean} Nt + (1 - P_f) P_{mean} t' + P_{error} (1 - P_{mean}) Nt + (1 - P_{error}) (1 - P_{mean}) t') + (M + \Delta M)(t_s + t_c) \quad (7)$$

其中, M 为由式(1)计算得到的抽样数量; ΔM 为分别由式(3)和式(1)算得 M 的差; $P_{mean} = (1 - \varepsilon)^m$ 为抽样中数据全为 inliers 的平均概率.

由式(7)和式(2), 相对 RANSAC 算法, PERANSAC 算法可以减少的计算时间为

$$\Delta T = (M + \Delta M)(1 - P_f P_{mean} - P_{error} (1 - P_{mean})) \Delta t - \Delta M(t_s + t_c + Nt) \quad (8)$$

其中, $1 - P_f P_{mean} - P_{error} (1 - P_{mean})$ 为模型不能通过预检验的概率. $\Delta t = Nt - t'$ 为一个模型参数进行全数据检验和进行预检验的时间差. 本文的预检验过程需要时间 nt , $\Delta t = (N - n)t$, 则式(8)变为

$$\Delta T = (M + \Delta M)(1 - P_f P_{mean} - P_{error} (1 - P_{mean}))(N - n)t - \Delta M(t_s + t_c + Nt) \quad (9)$$

从式(9)可以看出, 影响 ΔT 的因素很多, 对于确定的原始数据和模型, P_{mean} , t_s , t_c , t , N , n 都是确定的值, 影响 ΔT 的主要因素为正确模型参数和错误模型参数通过预检验的概率 $P_f, P_{error}, P_{error}$ 是未知的, 且对不同的原始数据是不同的, 对于相同的原始数据, P_{error} 会随着 P_f 的增加而增加. 由式(4)可知, 正确模型参数通过率 P_f 和数据错误率 ε 直接相关. 但在实际计算时数据错误率 ε 往往是未知的, 只能估计一个错误率 ε , ε 估计的是否合适直接影响正确模型参数通过率 P_f 和错误模型参数通过率 P_{error} , 进而影响计算效率. 实验中, 我们对真实图像数据在不同估计错误率下的计算效率进行分析.

一个好的预检验过程, 要求好的模型参数通过率 P_f 要高, 而错误模型参数 P_{error} 通过率很低. 事实上, 当好的模型参数通过率 P_f 比较高时, 设置的门限较低, 往往会造成错误模型参数通过率 P_{error} 变高, 在进一步的研究中会讨论如何设计更好的预检验过程.

2.4 算法步骤

- 1) 由式(4), 根据预检验最小通过概率和数据错误率, 以及参与预检验的样本数 n , 计算 n_f 和通过概率 ε ;
- 2) 由式(3), 根据通过概率 P_f , 置信概率 P 和估计错误率 ε 来计算需要选择的抽样数量 M ;
- 3) 随机抽取一个抽样, 并计算抽样对应的模型参数;
- 4) 随机选择 n 个数据, 对模型参数进行预检验, 重复 3), 直到所有模型参数都检验完毕. 如果所有的模型参数都没有通过预检验, 则重新从 1) 开始进行计算. 如果连续多次出现没有模型参数能通过预检验, 则调高估计错误率, 重新计算;
- 5) 用全部原始数据检验通过预检验的模型参数, 获得每个模型参数的 inliers 数量;
- 6) 根据 inliers 数量和误差方差来选择最优的模型参数;
- 7) 用最优模型参数对应的 inliers 计算最终模型参数.

3 实验

实验通过计算同一场景不同视角图像间的基础矩阵来对比分析本文算法和 RANSAC 算法; 由于算法带有随机性, 为了更真实地反映算法的特性, 实验给出的结果都是 500 次计算结果的平均值.

3.1 模拟数据实验

实验模拟同一场景不同视角两幅图像上的匹配点^[9], 匹配点数为 100. 为了模拟真实的图像特征点匹配效果, 在每一个模拟特征点上叠加方差为 2 个像素的高斯噪声.

实验参数: 置信概率 $P = 99\%$; 预检验样本数 $n = 15$; 预检验通过率 $P_f > 80\%$; 数据错误率 = 模拟数据的错误率. 实验在不同错误率下对比了 RANSAC 算法和 PERANSAC 算法的计算精度和计算效率. 图 1 中横轴为数据错误率, 纵轴为左右视图平均对极距离, 它反映了基础矩阵求解的精度. 从图中可以看出, PERANSAC 算法和

RANSAC 算法在不同的数据错误率下的计算精度保持一致,证明了本文算法在计算精度上的处理是正确的.图 2 给出在不同错误率下两种算法计算所用时间;图 3 给出 PERANSAC 算法相对 RANSAC 算法的速度提高程度,相对提高程度在 30%~50%之间.从计算结果可以看出,本文的算法相对于 RANSAC 算法,在计算效率上有当程度的提高.

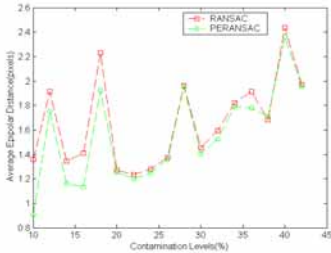


Fig.1 The average epipolar distance at different contamination levels
图 1 不同数据错误率下算法的平均对极距离

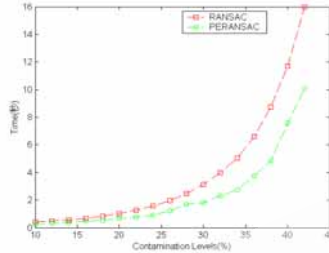


Fig.2 Time at different contamination levels
图 2 不同数据错误率下算法的计算时间

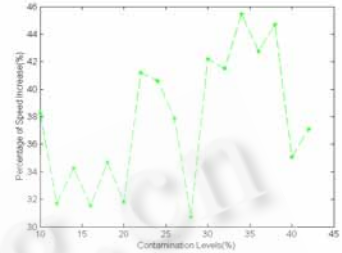


Fig.3 Percentage of speed increase at different contamination levels
图 3 不同数据错误率下算法的速度提高百分比

3.2 真实图像实验

3.2.1 场景 1 实验结果

实验参数:左右视图图像(如图 4 所示)匹配点数为 250;置信概率 $P = 99\%$; 预检验样本数 $n = 15$;预检验通过率 $P_f > 80\%$.



Fig.4 The images of left and right views of scene one
图 4 场景 1 左右视图图像

图 5 为不同估计错误率下 RANSAC 和 PERANSAC 算法的计算精度.可以看出,PERANSAC 算法的计算精度要优于 RANSAC 算法,特别是在估计错误率比较低的情况下.这是因为估计错误率偏低,会造成预检验的门槛偏高.对能通过预检验的模型参数质量要求比较严,算法在没有模型参数能够通过预检验时,重新抽样,重新计算.重新抽样相当于在原来抽样的基础上增加可供选择的抽样数量,而且每次增加的数量等于原来的抽样数;这样,最终选择的模型参数等价于在高估计错误率下,从更多的抽样中筛选出最优的,所以,计算结果优于 RANSAC 算法.

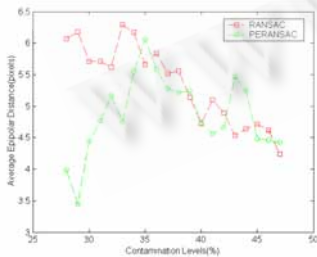


Fig.5 The average epipolar distance at different contamination levels
图 5 不同估计错误率下算法的平均对极距离

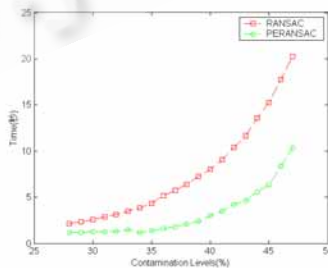


Fig.6 Time at different contamination levels
图 6 不同估计错误率下算法的计算时间

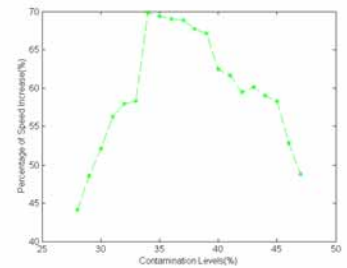


Fig.7 Percentage of speed increase at different contamination levels
图 7 不同估计错误率下算法的速度提高百分比

图 6、图 7 分别给出在不同估计错误率下 RANSAC 和 PERANSAC 算法计算需要的时间以及 PERANSAC 相对 RANSAC 算法在速度上提高的程度.从图中看出,PERANSAC 算法相对 RANSAC 算法速度提高程度高达 70%.算法速度提高的程度随估计错误率的提高而降低,这是因为估计错误率高于图像数据的真实错误率,抽样中数据全为 inliers 的样本增多.同时,利用估计错误率计算的预检验门限偏低,造成大量模型参数能通过预检验,影响了算法的整体效率.当估计错误率过低时,需要重新抽样,重新计算,影响了算法的整体效率.

3.2.2 场景 2 实验结果

图像左右视图(如图 8 所示)的匹配点数为 268,其他实验参数同场景 1 的实验.类似场景 1 的计算结果.



Fig.8 Images of left and right views of scene two

图 8 场景 2 左右视图图像

图 9~图 11 分别给出 PERANSAC 算法和 RANSAC 算法的对比结果.

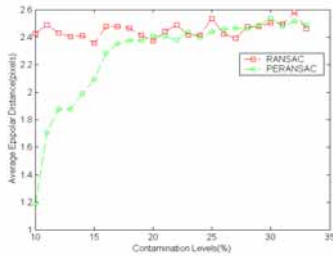


Fig.9 The average epipolar distance at different contamination levels

图 9 不同估计错误率下算法的平均对极距离

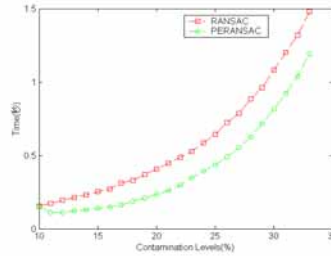


Fig.10 Time at different contamination levels

图 10 不同估计错误率下算法的计算时间

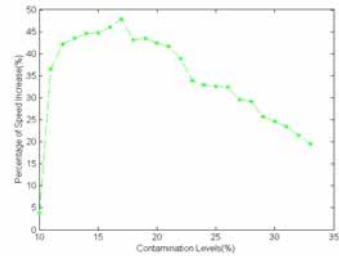


Fig.11 Percentage of speed increase at different contamination levels

图 11 不同估计错误率下算法的速度提高百分比

4 结 论

本文在 RANSAC 算法基础上,提出了一种基于模型参数预检验的 RANSAC 算法(PERANSAC),算法在 RANSAC 算法的模型参数检验之前,加入一个预检验过程,在不改变计算结果置信度的前提下,过滤掉大量偏差较大的模型参数,减少了模型参数检验的计算量,提高了 RANSAC 算法的整体效率.文中给出了 PERANSAC 算法的理论基础以及具体的算法步骤,并从理论上初步给出 PERANSAC 算法可以减少的计算量.分别用模拟数据和真实图像对本文算法进行了实验验证.结果证明,本文算法的计算精度和 RANSAC 算法精度保持一致,计算速度大大高于 RANSAC 算法.文中用小样本数据中 inliers 数量作为预检验标准,讨论了不同估计错误率对计算的影响.在以后的研究中,将讨论其他的预检验方法,以及不同的预检验方法对算法效率影响.并在理论上讨论算法效率和各个计算参数的关系.

致谢 在此,我们向对本文的工作给予支持和建议的同学,尤其是郭武同学表示感谢.

References:

- [1] Brandt S. Maximum likelihood robust regression with known and unknown residual models. In: Proc. of the ECCV 2002. 2002. 97-102.
- [2] Murray PTD. The development and comparison of robust methods for estimating the fundamental matrix. Int'l Journal of Computer Vision, 1996. 1-33.

- [3] Zhang ZY. Determining the epipolar geometry and its uncertainty: A review. *Int'l Journal of Computer Vision*, 1998,27(2): 161-195.
- [4] Fischler MA, Bolles RC. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 1981,24(6):381-395.
- [5] Chen ZZ, Wu CK. A linear algorithm with high accuracy for estimating fundamental matrix. *Journal of Software*, 2002,13(4): 840-845 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/13/840.pdf>
- [6] Rousseeuw PJ. *Robust Regression and Outlier Detection*. New York: John Wiley & Sons, 1987.
- [7] Torr PHS, Murray DW. Outlier detection and motion segmentation. *SPIE* 93, 1993. 432-443.
- [8] Stewart CV. MINPRAN: A new robust operator for computer vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1995,17(10):925-938.
- [9] Torr PHS, Zisserman A. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 2000,78:138-156.

附中文参考文献:

- [5] 陈泽志,吴成柯.一种高精度估计的基础矩阵的线性算法. *软件学报*,2002,13(4):840-845. <http://www.jos.org.cn/1000-9825/13/840.pdf>

敬告作者

《软件学报》创刊以来,蒙国内外学术界厚爱,收到许多高质量的稿件,其中不少在发表后读者反映良好,认为本刊保持了较高的学术水平.但也有些稿件因不符合本刊的要求而未能通过审稿.为了帮助广大作者尽快地把他们的优秀研究成果发表在我刊上,特此列举一些审稿过程中经常遇到的问题,请作者投稿时尽量予以避免,以利大作的发表.

1. 读书偶有所得,即匆忙成文,未曾注意该领域或该研究课题国内外近年来的发展情况,不引用和不比较最近文献中的同类结果,有的甚至完全不列参考文献.

2. 做了一个软件系统,详尽描述该系统的各个方面,如像工作报告,但采用的基本上是成熟技术,未与国内外同类系统比较,没有指出该系统在技术上哪几点比别人先进,为什么先进.一般来说,技术上没有创新的软件系统是没有发表价值的.

3. 提出一个新的算法,认为该算法优越,但既未从数学上证明比现有的其他算法好(例如降低复杂性),也没有用实验数据来进行对比,难以令人信服.

4. 提出一个大型软件系统的总体设想,但很粗糙,而且还没有(哪怕是部分的)实现,很难证明该设想是现实的、可行的、先进的.

5. 介绍一个现有的软件开发方法,或一个现有软件产品的结构(非作者本人开发,往往是引进的,或公司产品),甚至某一软件的使用方法.本刊不登载高级科普文章,不支持在论文中引进广告色彩.

6. 提出对软件开发或软件产业的某种观点,泛泛而论,技术含量少.本刊目前暂不开办软件论坛,只发表学术文章,但也欢迎材料丰富,反映现代软件理论或技术发展,并含有作者精辟见解的某一领域的综述文章.

7. 介绍作者做的把软件技术应用于某个领域的工作,但其中软件技术含量太少,甚至微不足道,大部分内容是其他专业领域的技术细节,这类文章宜改投其他专业刊物.

8. 其主要内容已经在其他正式学术刊物上或在正式出版物中发表过的文章,一稿多投的文章,经退稿后未作本质修改换名重投的文章.

本刊热情欢迎国内外科技界对《软件学报》踊跃投稿.为了和大家一起办好本刊,特提出以上各点敬告作者.并且欢迎广大作者和读者对本刊的各个方面,尤其是对论文的质量多多提出批评建议.