

文章编号:1001-9081(2010)02-0427-03

## 硬实时调度的线性近似分析方法

莫铁强,段 珊

(湖南涉外经济学院 计算机科学与技术系,长沙 410205)

(mtq12345@163.com)

**摘 要:**通过线性逼近硬实时系统任务的工作负荷量的方法,一个更加接近精确响应时间的上限能有效地降低调度分析时间。同时该上限用于任务集的充分性可调度测试时具有线性时间的复杂度。这种线性上限的可调度性测试能够用于交互的系统工具设计、基于搜索的系统优化以及任务集的动态接纳新任务的设计中。并且新的调度系统模型无时间死线、抖动大小限制,适用范围更广。相关的实验也表明响应时间上限可调度性分析提高了准确调度测试的效率。

**关键词:**繁忙周期;硬实时;线性近似

**中图分类号:** TP316.2 **文献标志码:** A

## Scheduling analysis of linear approximate response time in hard real-time system

MO Tie-qiang, DUAN Shan

(Department of Computer Science, Hunan International Economics University, Changsha Hunan 410205, China)

**Abstract:** By the approaches of being linearly close to the workload of tasks in hard real-time system, the closer response time upper bound can be used to decrease the running time. At the same time, it is linear in time of complexity for the upper bound in the sufficient schedulable tests of task sets. In interactive system design tools, system optimization via search, and admission of new tasks into dynamic systems, this new linear upper bound can considerably improve the efficiency of test. Further, the scheduling model has wide application range because it does not have the limits of deadline and jitter. The pertinent experiments show that the method of utilizing the approximate upper time improves the efficiency of accurate test of the scheduling.

**Key words:** busy period; hard real-time; linear approximation

### 0 引言

固定优先级调度机制在各种实时系统如汽车电子、数字机顶盒和医疗系统中广泛地使用,从而该调度机制目前已被各种商业实时操作系统支持。在这种调度机制中,通常所说的响应时间分析是不断计算每个任务的最差响应时间(Worst-case Response Time)以判断该时间是否在其时间死线之内,如果在其所对应的的时间死线之内,则该任务是可调度的。这种具有伪多项式时间复杂性<sup>[1]</sup>的响应时间分析及其各种有效的实现机制<sup>[2-3]</sup>能在合理的时间之内完成任务的调度分析,但在某些应用环境如交互式系统设计工具<sup>[4]</sup>、基于空间搜索的系统优化设计<sup>[5]</sup>等,任务可调度性分析的频繁调用所带来的时间开销已成为一种不可忽略的限制性因素,因此研究近似的多项式时间调度分析是一种必然的选择。

Fisher等人<sup>[6]</sup>早在2005年就提出了用部分线性递归方程的方法来完成线性时间的求解,这种方法在文献[7]中进一步扩展到任务激活时刻具有抖动(jitter)的调度模型中。一种具有更好的任务响应时间上限逼近被Bini和Baruah<sup>[8]</sup>提出,这种响应时间适用于无激活时刻抖动且具有任意时间死线的抢占任务集调度模型。最近一种针对任务到达具有先后关系的调度模型的近似方法在文献[9]中完成了高效的实现。

本文通过一个线性函数近似逼近任务的工作负荷量,然后计算所有高优先级任务对某个较低优先级任务的线性干扰之和,进一步计算level- $i$ 繁忙周期。我们的近似方法中允许出现任意时间死线、任意激活时刻抖动以及因任务的资源

共享而呈现的任务执行阻塞。相关的实验也表明了我们的线性近似方法能显著提高可调度性测试的效率。

### 1 任务调度模型

我们假设系统由 $n$ 个静态的任务组成( $\tau_1, \dots, \tau_n$ )且每个任务依次分配从1到 $n$ 的优先级别( $n$ 为具有最小的优先级别)。

每个任务 $\tau_i$ 的参数可由一个四元组 $\langle C_i, D_i, T_i, J_i \rangle$ 来描述,其中的参数依次是最差执行时间、相对时间死线、最小到达间隔时间和激活时刻抖动大小,每个任务的利用系数 $U_i$ 定义为 $C_i/T_i$ 。在我们的模型中允许任意的抖动大小和任意的时间死线大小( $D_i > T_i$ 或 $D_i < T_i$ )。

为了计算任务的最差响应时间,所有的任务的第1次激活时刻可以相同,这一时刻通常称为临界时刻<sup>[10]</sup>,同时把这种任务集称为同步任务集,相反带有到达时间偏移量的任务集则称为异步任务集。本文仅考虑同步任务集的近似响应时间计算。任务在访问互斥资源时可能被较低优先级的任务阻塞而造成其最大的执行延迟记为 $B_i$ 。任务的最差响应时间 $R_i$ 为从任务激活到执行结束的最长时间长度,如果 $R_i$ 满足不等式 $R_i \leq D_i - J_i$ ,则称该任务是可调度的。任务的level- $i$ 繁忙周期 $W_i$ 定义为从临界时刻开始到所有优先级不低于 $i$ 的任务的执行结束的时间长度。

### 2 响应时间分析

正如上文所述,对于同步任务集,任务 $\tau_i$ 与优先级别高于它的所有任务同时激活才会使任务 $\tau_i$ 出现最差响应时间,

收稿日期:2009-08-19。

作者简介:莫铁强(1971-),男,湖南桃江人,讲师,博士,主要研究方向:实时调度;段珊(1969-),女,湖南岳阳人,副教授,硕士,主要研究方向:嵌入式系统。

也将导致出现最长的  $level-i$  繁忙周期。下面我们将分时间死线大于周期和小于周期计算任务的响应时间。

### 2.1 时间死线小于或等于周期的抢占调度分析<sup>[3]</sup>

对于时间死线小于或等于周期的抢占调度机制,任务  $\tau_i$  的每次调用需在该任务的下次调用激活之前执行结束。因此任何  $level-i$  繁忙周期最多含有任务  $\tau_i$  的一次调用,也就是任务  $\tau_i$  的最差响应时间等于该任务的  $level-i$  繁忙周期  $W_i$ 。具体地说,可用下面的公式加以描述:

$$R_i = W_i = B_i + C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{W_i + J_j}{T_j} \right\rceil C_j \quad (1)$$

该公式中符号  $\lceil a \rceil$  表示对  $a$  求不小于  $a$  的最小整数,  $hp(i)$  表示任务集中所有优先级别高于任务  $\tau_i$  的任务子集。从该公式中可以发现后面的求和部分表示了所有优先级别高于任务  $\tau_i$  的任务对当前任务的执行造成的干扰而引起当前任务推迟执行。

由于在方程(1)中  $W_i$  出现在方程的两边且  $W_i$  是单调非减的函数,故对  $W_i$  的求解可用  $W_i^0 = B_i + C_i$  作为初始值进行不动点计算直至  $W_i^m = W_i^{m+1}$ 。这样当  $W_i^m > D_i - J_i$  时,任务  $\tau_i$  不可调度。

### 2.2 具有任意时间死线的调度分析<sup>[3]</sup>

对于这种具有任意时间死线的任务集,某个任务的一次调用可能在它的下一次调用激活之前并未执行结束,这样在任务  $\tau_i$  的  $level-i$  繁忙周期内可能存在该任务的多次调用执行。从而对于任务的最差响应时间的计算,需要考虑在该任务的繁忙周期内当前任务的调用次数。开始于临界时刻结束于当前任务  $\tau_i$  的第  $q$  次调用的繁忙周期可通过下方方程的不动点的计算来完成( $q=0$  为任务  $\tau_i$  的第1次调用):

$$W_i^{m+1}(q) = B_i + (q+1)C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{W_i^m + J_j}{T_j} \right\rceil C_j \quad (2)$$

递归计算的初始值为  $W_i^0(q) = B_i + (q+1)C_i$ , 任务的第  $q+1$  次调用的最差响应时间  $R_i(q)$  为  $W_i^{m+1}(q) - qT_i$ , 这样当  $W_i^{m+1}(q) - qT_i > D_i - J_i$  时,任务  $\tau_i$  的第  $q+1$  次调用不可调度。

如果任务  $\tau_i$  的第  $q+1$  次调用的结束时刻迟于其后续调用的激活时刻,则该任务的第  $q+1$  次调用必然推迟后续调用的执行。从而我们需要计算调用  $q=0,1,2,\dots$  直到第  $q+1$  次调用在其下次调用最早可能的激活时刻之前执行结束,也就是  $W_i(q) \leq (q+1)T_i - J_i$ 。这样任务  $\tau_i$  的最差响应时间  $R_i$  为  $\forall q, \max\{W_i(q) - qT_i, \dots\}$ , 且只要满足  $R_i \leq D_i - J_i$ , 任务  $\tau_i$  就可调度。

## 3 响应时间上限分析

在这部分我们将分析任意时间死线和任意激活时刻抖动大小的固定优先级调度的响应时间上限大小。类似于文献[8]中的方法,在任何时间间隔内,我们从执行高优先级别任务  $\tau_j$  的时间中推导出与这一时间逼近的线性上限函数,然后由这个函数计算任务的响应时间上限。

假设较高优先级任务  $\tau_j$  在系统最差执行时的  $[0, t]$  时间间隔内所需要的最大执行时间为  $w_j(t)$ , 而当只有该任务运行于处理器上时其最大执行时间为  $w_j^m(t)$ 。因为有不同任务运行于处理器上时彼此会干扰,故  $\forall j, \forall t, w_j^m(t) \geq w_j(t)$ 。当只有任务  $\tau_j$  运行于处理器上时,在抖动大小为  $J_j$  的时间间隔内共有  $\lfloor J_j/T_j \rfloor + 1$  次 ( $\lfloor a \rfloor$  表示对  $a$  求不大于  $a$  的最大整数) 该任务的激活并连续执行,在这些调用执行结束前,仍将有新的该任务调用的激活,也就是处理器首先执行  $h$  次  $\tau_j$  任务调用然后空闲一段时间,紧接着每  $T_j$  内,执行  $C_j$  大小的  $\tau_j$  任务,

如图1所示,在图中的线性上限约束的斜率为  $U_j = C_j/T_j$ 。这样上限约束  $w_j^{ub}(t)$  满足  $\forall t, w_j^{ub}(t) \geq w_j^m(t)$ , 图中的点  $p(t, y)$  表示当在某个最小的  $t$  时,  $w_j^{ub}(t) = w_j^m(t)$ 。为了确定线性上限约束函数,我们就需要确定  $h$  值和  $p(t, y)$  的坐标值。

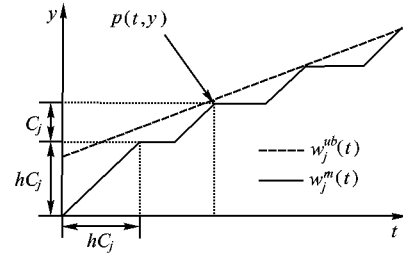


图1 线性近似图解

在  $\lfloor J_j/T_j \rfloor + 1$  次任务  $\tau_j$  的激活后,紧接着的任务激活时刻发生在

$$(\lfloor J_j/T_j \rfloor + 1)T_j - J_j + (k-1)T_j; k=1,2,\dots \quad (3)$$

由于连续执行  $h$  次  $\tau_j$  任务的时间长度必然大于或等于式(3)中的激活时刻,也就是  $(\lfloor J_j/T_j \rfloor + 1)C_j + (k-1)C_j \geq (\lfloor J_j/T_j \rfloor + 1)T_j - J_j + (k-1)T_j$ , 从而我们有  $k \leq J_j/(T_j - C_j) - \lfloor J_j/T_j \rfloor$ , 取  $k$  的最大值即:

$$k = \lfloor J_j/(T_j - C_j) \rfloor - \lfloor J_j/T_j \rfloor \quad (4)$$

这样连续执行的次数  $h$  满足  $h = k + (\lfloor J_j/T_j \rfloor + 1) = \lfloor J_j/(T_j - C_j) \rfloor + 1$ 。点  $P(t, y)$  中  $y = hC_j + C_j$ ,  $t$  为连续执行的结束时刻加上  $T_j$ , 即对式(3)中  $k$  用式(4)中最大值替换并加上  $C_j$  和  $T_j$  后得到  $t = (\lfloor J_j/(T_j - C_j) \rfloor + 1)T_j - J_j + C_j$ 。根据上限线性约束  $w_j^{ub}(t)$  的斜率和坐标  $p(t, y)$ , 我们有:

$$w_j^{ub}(t) = U_j t + U_j J_j + C_j(1 - U_j) \quad (5)$$

根据式(5)所有优先级别高于  $i$  的任务导致  $\tau_i$  延迟执行的干扰上限总和为  $\sum_{\forall j \in hp(i)} w_j^{ub}(t) = t \sum_{\forall j \in hp(i)} U_j + \sum_{\forall j \in hp(i)} (U_j J_j + C_j(1 - U_j))$ 。所以处理器上执行当前任务  $\tau_i$  时间  $C$  和所有较高优先任务的时间开销上限值为直线  $y = t$  与直线  $y = C + t \sum_{\forall j \in hp(i)} U_j + \sum_{\forall j \in hp(i)} (U_j J_j + C_j(1 - U_j))$  的交集, 记为:

$$O_i^{ub}(C) = \frac{C + \sum_{\forall j \in hp(i)} (U_j J_j + C_j(1 - U_j))}{1 - \sum_{\forall j \in hp(i)} U_j} \quad (6)$$

进一步当  $C = B_i + (q+1)C_i$ , 我们得到  $level-i$  繁忙周期大小的上限为:

$$W_i^{ub}(q) = \frac{B_i + (q+1)C_i + \sum_{\forall j \in hp(i)} (U_j J_j + C_j(1 - U_j))}{1 - \sum_{\forall j \in hp(i)} U_j} \quad (7)$$

这样我们有  $\tau_i$  任务的第  $q+1$  次调用的响应时间上限:

$$R_i^{ub}(q) = w_i^{ub}(t) - qT_i \quad (8)$$

以及该任务的响应时间上限:

$$R_i^{ub} = \max_{\forall q} (W_i^{ub}(q) - qT_i) \quad (9)$$

由式(8)有  $R_i^{ub}(q) - R_i^{ub}(q+1) = T_i - \frac{C_i}{1 - \sum_{\forall j \in hp(i)} U_j} \geq 0$  (因

为  $\sum_{i=1}^i \frac{C_i}{T_i} \leq \sum_{i=1}^n \frac{C_i}{T_i} \leq 1$ ), 故  $R_i^{ub}(q)$  是  $q$  的单调非增函数, 从而  $R_i^{ub}(q)$  的最大值即响应时间上限为:

$$R_i^{ub} = R_i^{ub}(0) = W_i^{ub}(q) = \left[ B_i + C_i + \sum_{\forall j \in hp(i)} (U_j J_j + C_j(1 - U_j)) \right] /$$

$$\left(1 - \sum_{j \in hp(i)} U_j\right) \quad (10)$$

显然对该上限值令因低优先级任务阻塞而引起的当前任务延迟执行的时间大小为  $B_i = 0$  以及激活时刻的抖动大小  $J_i = 0$ , 我们将得到文献[8]中的上限值。

从式(10)中我们可得到任务  $\tau_i$  可调度性的一个充分条件:  $R_i^{ub} < D_i - J_i$ 。在检测任务集的可调度性时,按照优先顺序由大到小依次进行检测,每次可调度性检测的时间复杂度为  $O(i)$ ,并且每次进行上限计算时可以利用上次已求出的和值,故任务集可调度性检测的时间复杂度为  $O(n)$ 。

#### 4 实验分析

在这部分我们将通过实验来说明本文的最差响应时间的线性近似上限分析方法的有效性。在每次实验中,我们将利用对统计结果不产生偏倚的算法<sup>[10]</sup>来构造任务集合并与 Liu 和 Layland 的基于任务利用系数上限<sup>[11]</sup>、Fisher 和 Baruah 的上限<sup>[6]</sup>以及式(2)中所描述的精确调度方法进行相关调度性能的比较。

##### 4.1 任务集的形成

在任务集的  $n$  个任务中,按周期  $T$  的 10 的  $L$  ( $L$  为正整数)次幂进行任务集合划分,即当  $T \in [10^t, 10^{t+1}]$  ( $t = 0, 1, 2, \dots, L-1$ ) 的任务数为  $n/L$ ,然后任务的周期在相应的区间上均匀分布。这种分配方法在目前商业实时操作系统的测试中称为复制周期分布类型。对于每个任务的利用系数我们采用对调度机制无任何偏倚的 UUnifast 算法<sup>[10]</sup>来产生  $U_i$ ,这样可以确定参数执行时间  $C_i = U_i T_i$ 。另外,除了另有说明实验使用下面的默认参数:任务集大小  $n = 25$ ,  $D_i = T_i$ ,  $B_i = J_i = 0$ ,  $L = 2$ ,  $U = 60\%$ ,图中所使用的任务集数目为 5000。

##### 4.2 实验

**实验 1** 此实验将说明我们的线性响应时间上限约束的可调度性效率与任务集的利用系数以及任务周期分布的关系。图 2 中接收比例 (Acceptance ratio) 用来描述可调度性测试的效率,它定义为采用上限约束得到的可调度任务集数目与采用式(2)中所描述的能准确调度的任务集数目的比值。图 2 说明了上限约束的调度性能与任务周期的分布有关,当  $L \geq 2$  时,响应时间上限的可调度性性能显然好于文献[11]的可调度性测试的性能,但当  $L = 1$  时却相反。

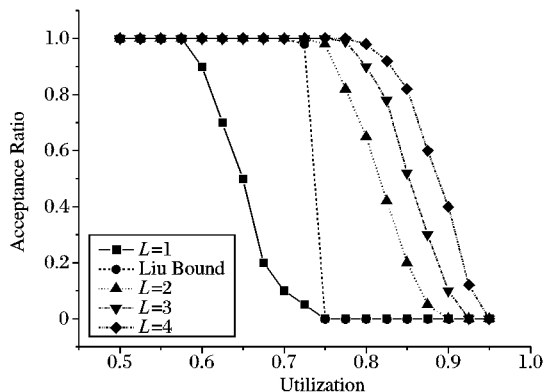


图2 调度效率与任务利用系数关系

**实验 2** 此实验将说明不同的上限约束的可调度性效率与时间死线和周期的比值(从 0.05 到 0.95)的关系。图 3 中任务集可调度比例为某种上限约束可调度测试所形成的可调度任务集合数量与产生的任务集合总数的比值。图 3 表明当  $D_i \geq 0.3T_i$  时,产生的任务集在精确调度下是可调度的,而我们的线性上限约束却在  $D_i \geq 0.45T_i$  时,产生的任务集大多数

是可调度的。

**实验 3** 此实验将说明不同上限约束的可调度性效率与任务激活时刻抖动大小的关系。图 4 中横轴为抖动大小与任务周期的比值,其他参数使用默认的值:  $n = 25$ ,  $D_i = T_i$ ,  $B_i = 0$ ,  $L = 2$ ,  $U = 60\%$ ,该图表明当  $J_i \leq 0.5T_i$  时,精确调度测试能保证大多数任务是可调度的,而我们的线性上限约束在  $J_i \leq 0.35T_i$  能保证大多数任务的可调度性,Liu 等人基于利用系数上限需在  $J_i \leq 0.2T_i$  才能保证大多数任务的可调度性。同时该实验也表明了上限约束可调度性测试对任务激活时刻存在相对该任务的周期具有较小的抖动时仍具有一定的有效性。

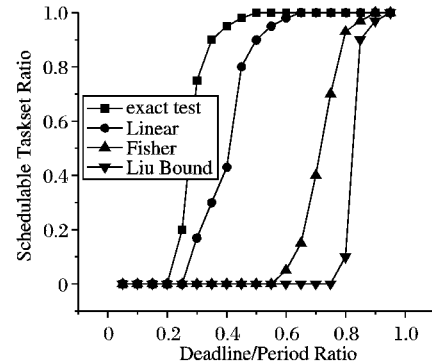


图3 调度效率与死/周期比例关系

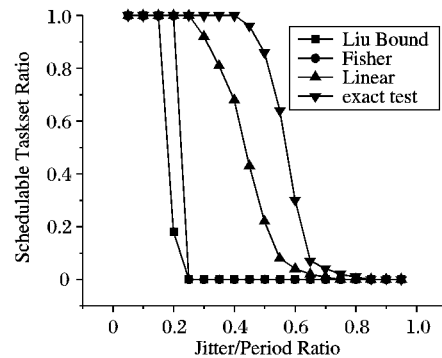


图4 调度效率与抖动/周期比例关系

#### 5 结语

本文通过详细的响应时间分析得到了一个更加逼近的响应时间上限,该上限用于任务集的充分性可调度测试时具有线性时间的复杂度。这种线性上限的可调度性测试能够用于交互的系统工具设计、基于搜索的系统优化以及任务集的动态接纳新任务的设计中;并且我们的系统模型无时间死线、抖动大小限制,适用范围更广。相关的实验也表明我们的响应时间上限可调度性分析提高了准确调度测试的效率。

##### 参考文献:

- [1] EISENBRAND F, ROTHVOB T. Static priority real-time scheduling: Response time computation is NP-hard[C]// Proceedings of the 2008 Real-time Systems Symposium. Washington, DC: IEEE Computer Society, 2008: 397-406.
- [2] BRIL R J, VERHAEGH W F, POL E-J D. Initial values for on-line response time calculations[C]// Proceedings of the 15th Euro Conference on Real-time Systems. [S.l.]: IEEE, 2003: 13-22.
- [3] DAVIS R I, ZABOS A, BURNS A. Efficient exact schedulability tests for fixed priority pre-emptive systems[J]. IEEE Transactions on Computers, 2008, 57(9): 1261-1276.
- [4] RACU R, HAMANN A, ERNST R. Sensitivity analysis of complex embedded real-time systems[J]. Real-time Systems, 2008, 39(1/3): 31-72.

加速搜索功能和筛选功能和图像金字塔搜索策略的使用,极大地约束了模板匹配和最小二乘法计算量,从而使得匹配速度很快;而形状匹配和模板匹配又保证了匹配的稳健性;最小

二乘法能够有效提高匹配精度和保证精度的稳定性。实验证明该算法是行之有效的,为光刻技术提供了一种精确识别定位匹配方法。

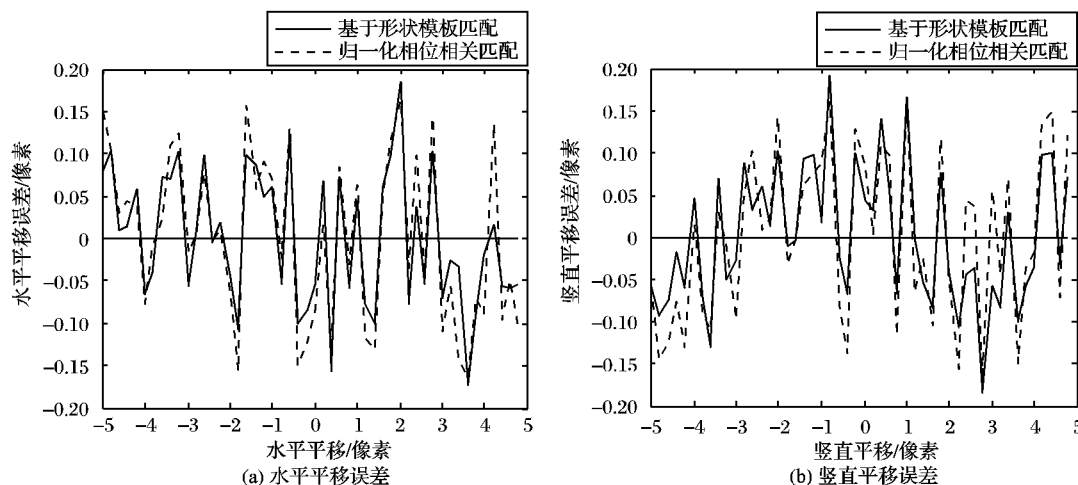


图5 平移精度比较

#### 参考文献:

- [1] 刘进, 闫利. 图像相关匹配算法的快速实现[J]. 武汉大学学报: 信息科学版, 2007, 32(8): 684-687.
- [2] 罗钧, 侯艳, 付丽. 一种改进的灰度矩边缘检测算法[J]. 重庆大学学报, 2008, 31(5): 549-552.
- [3] PETER A, RANGARAJAN A. Shape analysis using the Fisher-Rao Riemannian metric: Unifying shape representation and deformation [C]// Proceedings of the 3rd IEEE International Symposium on Biomedical Imaging: Nano to Macro. Washington, DC: IEEE Computer Society, 2006: 1164-1167.
- [4] PETER A, RANGARAJAN A. Maximum likelihood wavelet density estimation with applications to image and shape matching[J]. IEEE Transactions on Image Processing, 2008, 17(4): 458-468.
- [5] EL MUNIM H A, FARAG A A. Shape representation and registration using vector distance functions [C]// CVPR'07: Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition. Washington, DC: IEEE Computer Society, 2007: 1-8.
- [6] 张素, 苏和, 陆雪松, 等. 基于形状匹配的快速图像配准[J]. 天津大学学报, 2008, 41(4): 433-438.
- [7] STEGER C. Occlusion, clutter, and illumination invariant object recognition [J/OL]. International Archives of Photogrammetry and Remote Sensing, 2002, XXXIV(3A): 345-350. [2009-08-01]. <http://www.isprs.org/commission3/proceedings02/papers/paper006.pdf>.
- [8] STEGER C. Similarity measures for occlusion, clutter, and illumination invariant object recognition [C]// Proceedings of the 23rd DAGM - Symposium on Pattern Recognition, LNCS 2191. Berlin: Springer-Verlag, 2001: 148-154.
- [9] ULRICH M, STERGER C. Performance comparison of 2D object recognition techniques [J/OL]. International Archives of Photogrammetry and Remote Sensing, 2002, XXXIV(3A): 368-374. [2009-08-01]. <http://www.isprs.org/commission3/proceedings02/papers/paper039.pdf>.
- [10] ULRICH M, STERGER C. Empirical performance evaluation of object recognition methods [C]// Empirical Evaluation Methods in Computer Vision. Los Alamitos: IEEE Computer Society Press, 2001: 62-76.
- [11] WALLACK A, MANOCHA D. Robust algorithms for object localization [J]. International Journal of Computer Vision, 1998, 27(3): 243-262.
- [12] STEGER C. Subpixel - precise extraction of lines and edges [J/OL]. International Archives of Photogrammetry and Remote Sensing, 2000, XXXIII(B3): 141-156. [2009-08-01]. <http://www9.in.tum.de/people/steger/publications/2000/ISPRS-Congress-00-Steger.pdf>.
- [13] TANIMOTO S L. Template matching in pyramids [J]. Computer Graphics and Image Processing, 1981, 16(4): 356-369.
- [14] BORGEFORS G. Hierarchical chamfer matching: A parametric edge matching algorithm [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1988, 10(6): 849-865.
- [15] 陈钊, 郭永彩, 许丹宇. 互相关 DPIV 亚像素匹配方法研究[J]. 应用力学学报, 2008, 25(2): 289-292.
- [16] DELMAS P, FARF G G, SHORIN A, et al. Robust least-squares image matching in the presence of outliers [C]// CAIP 2007: Proceedings of the 12th International Conference on Computer Analysis of Images and Patterns, LNCS 4673. Berlin: Springer, 2007: 776-783.
- [17] RICHARD P. Efficient computation of response time bounds under fixed-priority scheduling [C]// Proceedings of the 15th Conference on Real-time and Network Systems. [S.l.]: IEEE, 2007: 95-104.
- [18] MÄKI - TURJA J, NOLIN M. Efficient implementation of tight response-times for tasks with offsets [J]. Real-time Systems, 2008, 40(1): 73-113.
- [19] BINI E, BUTTAZZO G C. Measuring the performance of schedulability tests [J]. Real-time Systems, 2005, 30(1/2): 129-154.
- [20] LIU C L, LAYLAND J W. Scheduling algorithms for multiprogramming in a hard-real-time environment [J]. Journal of the ACM, 1973, 20(1): 46-61.

(上接第429页)

- [5] EMBERSON P, BATE I. Minimising task migration and priority changes in mode transitions [C]// Proceedings 13th IEEE RTAS 2007. Washington, DC: IEEE Computer Society, 2007: 158-167.
- [6] FISHER N, BARUAH S. A fully polynomial - time approximation scheme for feasibility analysis in static-priority systems with arbitrary relative deadlines [C]// Euromicro Conference on Real-time Systems. Amsterdam: IOS Press, 2005: 117-126.
- [7] RICHARD P, GOOSENS J. Approximating response times for static-priority tasks with release jitters [C/OL]// Euromicro Conference on Real-time Systems (WiP session). [2009-08-01]. [http://www.lisi.ensma.fr/ftp/pub/documents/papers/2006/2006-EICRTSW-](http://www.lisi.ensma.fr/ftp/pub/documents/papers/2006/2006-EICRTSW-Richard.pdf)