

一种实时多目标路径规划方法

魏 唯 欧阳丹彤 吕 帅

(吉林大学计算机科学与技术学院 长春 130012)

(吉林大学符号计算与知识工程教育部重点实验室 长春 130012)

摘 要 提出一种利用实时搜索思想的多目标路径规划方法。首先设计并实现局部路径规划算法,在有限的局部空间内执行启发式搜索,求解所有局部非支配路径;在此基础上,提出实时多目标路径规划方法,设计并实现相应的启发式搜索算法,在线交替执行局部搜索过程、学习过程与移动过程,分别用于求解局部空间内的最优移动路径,完成状态的转移和更新状态的启发信息,最终到达目标状态。研究表明,实时多目标启发式搜索算法通过限制局部搜索空间,避免了大量不必要的计算,提高了搜索效率,能够高效地求解多目标路径规划问题。

关键词 多目标路径规划,实时搜索,局部空间,启发式搜索

中图分类号 TP18 **文献标识码** A

Real-time Multiobjective Path Planning

WEI Wei OUYANG Dan-tong LV Shuai

(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

(Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China)

Abstract A method adopting the idea of real-time search for solving multiobjective path planning problems was proposed, a local path planning algorithm was designed and implemented which executes a heuristic search within a limited local space to get all of the local non-dominated paths. After that, the method of real-time multiobjective path planning was proposed and the corresponding heuristic search algorithm was designed and implemented. The algorithm executes path planning process, learning process and moving process online by turns to get the set of local optimal paths, transfers the current state in the local space and updates heuristic information of the local states respectively until reaching the goal state successfully. Test results show that the algorithm can solve multiobjective path planning problems efficiently by limiting local search space which can avoid lots of unnecessary computing work and thus improve the search efficiency.

Keywords Multiobjective path planning, Real-time search, Local space, Heuristic search

1 引言

路径规划问题中需要优化的目标函数超过一个并需要同时处理时,称为多目标路径规划。启发式搜索方法求解多目标路径规划问题取得了很好的成果。最早由 Stewart 与 White 提出利用启发式搜索方法求解多目标路径规划问题^[1]; Dasgupta 等人将非单调的启发信息应用于 VLSI 设计过程^[2]中; Mandow 等人提出了搜索过程中采用路径扩展的方法,提高了求解的准确性^[3]; Dasgupta 与 Harikumar 等人提出了用于解决多目标路径规划问题的深度优先搜索方法^[2,4]。状态空间启发式搜索规划器考虑了多个衡量尺度的折中^[5,6]。

很多实际问题对求解时间有严格的限制,特别是问题规模较大时,对整个问题一次性离线求解效率低下。实时搜索

的基本思想是在线执行规划求解过程,多次计算局部最优路径并沿所求路径进行移动,直至到达目标状态。最早由 Korf 提出实时启发式搜索算法 LRTA*, 其只向前观察深度为 1 的局部区域^[7]; Bulitko 与 Lee 提出的 LRTS 算法考虑了半径 d 区域内的所有状态^[8]。Koenig 与 Likhachev 提出的 RTAA* 算法采用了更容易实现的启发函数更新方法,使得学习速度更快^[9]。Rayner 等人提出的 P-LRTA* 算法通过衡量搜索空间中各个状态的重要性使得学习效率有明显的提高^[10]。

本文提出一种实时多目标启发式搜索算法 RTMHS。该算法并不是以离线状态对整个问题进行求解后再执行移动过程,而是以在线的方式交替执行路径规划过程、学习过程与移动过程。规划过程在当前局部空间内执行启发式搜索,只求出部分移动路径就开始进行移动过程,沿所选路径在局部空间内移动,向目标状态靠近。同时执行学习过程不断更新各

到稿日期:2009-08-05 返修日期:2009-10-19 本文受国家自然科学基金(60973089, 60773097, 60873044, 60873148), 教育部博士点基金(20060183044), 吉林省科技发展计划项目基金(20060532, 20080107)资助。

魏 唯(1984—),女,硕士生,主要研究方向为智能规划与自动推理, E-mail: weiwei_nenu@gmail.com; 欧阳丹彤(1968—),女,博士,教授,博士生导师, CCF 高级会员,主要研究方向为基于模型的诊断与自动推理; 吕 帅(1981—),男,博士生,主要研究方向为智能规划与自动推理。

个状态的启发信息,移动完毕之后再次执行规划过程,求解下一次移动的路径。如此反复执行,直到成功到达目标状态。

2 相关定义

多目标路径规划问题可以形式化地表示为三元组 $\langle G, S_0, \Gamma \rangle$, 其中 $G = (N, A, c)$ 表示状态空间图, N 为状态节点集合, A 为弧的集合, $c: A \rightarrow R^q$ 为转移代价函数, $c(m, n)$ 表示由状态节点 S_m 转移到状态节点 S_n 的 q 维代价向量, G 为无向图时, $c(m, n) = c(n, m)$; $S_0 \in N$ 表示初始状态; $\Gamma \in N$ 表示目标状态集合。

定义 1^[2] 设向量 f_1 和 f_2 为 K 维向量, 称 f_1 强支配 f_2 当且仅当 $\forall i, 1 \leq i \leq K$ 满足 $f_{1i} \leq f_{2i}$ 且 $f_1 \neq f_2$, 记作 $f_1 < f_2$, 其中 f_{1i} 和 f_{2i} 表示向量 f_1 和 f_2 的第 i 个元素。

定义 2^[2] 设向量 f_1 和 f_2 为 K 维向量, 称 f_1 弱支配 f_2 当且仅当 $\forall i, 1 \leq i \leq K$ 满足 $f_{1i} \leq f_{2i}$, 记作 $f_1 \leq f_2$, 其中 f_{1i} 和 f_{2i} 表示向量 f_1 和 f_2 的第 i 个元素。

定义 3^[3] 对于向量集合 X , 称向量 $x \in X$ 非支配当且仅当 $\exists y \in X$ 满足条件 $y < x$; 称 $nondom(X) = \{x \in X \mid \exists y \in X \text{ 满足 } y < x\}$ 为向量集合 X 的非支配向量集。

定义 4^[3] 设 P_{mn} 是状态 S_m 与状态 S_n 之间所有路径的集合, 路径 $p_1, p_2 \in P_{mn}$, 称 p_1 支配 p_2 当且仅当 $g_1 < g_2$, 记作 $p_1 < p_2$, 其中 g_1 与 g_2 分别为路径 p_1 与 p_2 的代价向量。

定义 5^[3] 设 P_{mn} 是状态 S_m 与状态 S_n 之间所有路径的集合, 路径 $p_1 \in P_{mn}$, 称 p_1 为非支配路径当且仅当 $\exists p_2 \in P_{mn}$ 满足 $p_2 < p_1$, 即 P_{mn} 中不存在其他路径在所有目标上都优于路径 p_1 。

多目标路径规划问题的最优路径集合为初始状态与目标状态之间的所有非支配路径的集合, 它们的特点是无法在改进任何目标函数的同时不削弱至少一个其他目标函数。

定义两种支配关系的原因如下: 1) 搜索过程中各个目标函数上权重都相等的两条路径具有同等的资格做进一步考虑, 此时要从所有候选路径中选出更有希望的路径进行扩展, 需要利用强支配关系; 2) 搜索过程中的一条路径的权重估值与已求出的一条解路径的权重相等时, 该路径扩展出的任何路径必然是被这条解路径强支配的, 因此该路径已没有进一步扩展的必要, 此时利用弱支配关系对搜索过程中的路径与解路径进行比较即可。

3 实时多目标启发式搜索算法

本节提出实时多目标启发式搜索算法, 分别描述算法的基本流程、局部搜索过程、学习过程与移动过程。

3.1 基本流程

下面首先给出实时多目标启发式搜索算法 (Real-time Multiobjective Heuristic Search, RTMHS) 的基本流程。

RTMHS($\langle G, S_0, \Gamma \rangle, d$)

1) 初始化: 令 $S_{cur} = S_0$;

2) 反复执行如下步骤, 直到 $S_{cur} \in \Gamma$:

a) 以 S_{cur} 为根节点, 执行深度为 d 的局部搜索过程, (LPATHS, CLOSED) = LMHS(S_{cur}, d);

b) 对 LPATHS 中的每条局部最优路径 p_m 执行学习过程 LEARNING($S_m, CLOSED$);

c) 从 LPATHS 中选择一条路径 p_u 作为局部空间内的移

动路径;

d) 执行向前移动过程, $S_{cur} = \text{MOVE}(S_{cur}, p_u)$ 。

算法从初始状态 S_0 出发, 局部搜索过程对当前状态周围的局部空间进行搜索, 得到局部空间内的最优解集。RTMHS 算法在当前状态下做深度为 d 的局部搜索, 得到长度为 d 的局部最优解集。执行一次局部搜索过程后, 学习过程修改局部空间中所有已扩展状态的启发信息。移动过程从所有局部最优路径中任选一条路径, 移动到局部空间的一个边界状态。

解决多目标路径规划问题需要求出所有非支配路径, 计算量较大, RTMHS 算法将问题的求解过程分解为多个局部求解过程。由于局部状态空间通常较小并且独立于整个状态空间, 因此局部搜索过程能够在较短的时间内完成, 可保证局部移动路径的最优性与问题求解的实时性, 以便尽快做出向前移动的决策。

3.2 局部搜索过程

局部搜索过程 (Local Multiobjective Heuristic Search, LMHS) 采用路径扩展方法, 每次扩展到到达某个状态节点一条路径, 利用路径的启发信息引导搜索过程, 以提高搜索效率。

定义 6 设 p_m 为当前局部搜索过程中从状态 S_{cur} 到达局部空间内状态 S_m 的一条路径, 称 $f_m = g_m + h_m$ 为路径 p_m 的代价估值, 其中当前状态 S_{cur} 作为当前局部搜索过程的初始状态, g_m 为路径 p_m 的代价向量, h_m 为路径 p_m 的启发信息, 估计每一个目标函数上从状态节点 S_m 到达目标状态的最小代价。

搜索过程分别用集合 $G_{op}(m)$ 和集合 $G_{cl}(m)$ 记录到达状态 S_m 的未扩展的路径和已扩展的路径; OPEN 表记录当前各个状态节点的所有未扩展路径, 每一个路径所对应的记录项为一个三元组 (S_m, g_m, f_m) ; CLOSED 用来记录局部空间中所有已扩展的状态。

定义 7 设 p_m 为 OPEN 表中一条未扩展路径, f_m 为路径 p_m 的代价估值, 称 p_m 为非支配 OPEN 路径当且仅当 $\exists (S_n, g_n, f_n) \in \text{OPEN}$ 使得 $f_n < f_m$ 。

局部搜索过程每一次循环利用向量之间的支配关系是从非空的 OPEN 表中选择一条非支配 OPEN 路径进行扩展, 并将扩展出的路径插入 OPEN 表, 当扩展到目标状态或局部空间的边界状态时, 求出一条局部路径。

定义 8 设 PATHS 为由当前状态 S_{cur} 到达局部空间边界状态的所有局部路径的集合, 称路径 $p_m \in \text{PATHS}$ 为一条局部非支配路径当且仅当 \exists 路径 $p_n \in \text{PATHS}$ 满足 $f_n < f_m$ 。

算法执行过程描述如下:

LMHS(S_{cur}, d)

1) OPEN = $\{(S_{cur}, g_{cur}, f_{cur})\}$, CLOSED 为空集;

2) 反复执行步骤 a) 一步骤 c), 直到 OPEN 表为空:

a) 从 OPEN 表中选择一条非支配的 OPEN 路径 p_m 进行扩展, 令 g_m 为该路径的代价向量, 将 p_m 所对应的 OPEN 表项 (S_m, g_m, f_m) 从 OPEN 表中删除, 并将 g_m 从 $G_{op}(m)$ 移到 $G_{cl}(m)$;

b) 若 $S_m \in \Gamma$ 或 $\text{deep}(S_m) = d$, 则 p_m 为一条局部最优路径, f_m 为路径 p_m 的代价, 将 p_m 加入 LPATHS, 并从 OPEN 表中删除所有其代价估值被 f_m 支配的路径;

c) 若 $\text{deep}(S_m) < d$, 则将 S_m 加入 CLOSED, 对 S_m 的所有

前驱状态 S_n , 生成到达 S_n 的路径 p_n , 令 $g_n = g_m + c(S_m, S_n)$, 若 g_n 不被 $G_{op}(n) \cup G_{cl}(n)$ 中任意路径代价支配, 则

i) 将 g_n 插入 $G_{op}(n)$, 并删除 $G_{op}(n) \cup G_{cl}(n)$ 中所有被 g_n 支配的路径;

ii) 计算路径 p_n 的代价估值 f_n , 若 f_n 不被 LPATHS 中任何路径的代价支配, 则将 (S_n, g_n, f_n) 插入 OPEN 表并记录相关的前驱信息;

3) 返回局部最优路径集合 LPATHS 与局部区域所有状态集合 CLOSED.

初始假设只有一条代价向量为零的路径到达初始状态 S_{cur} , $OPEN = \{S_{cur}, g_{cur}, f_{cur}\}$, CLOSED 为空集。当 OPEN 表为空时, 搜索过程终止, 输出所求得的最优路径集合; OPEN 表非空时, 从所有候选路径中, 根据路径的代价估值选择一条非支配的 OPEN 路径进行扩展。

对选出的一条路径 p_m 进行扩展时, 生成到达状态 S_m 的所有后继状态的路径。当生成的路径在所有到达该后继状态的路径中非支配时, 计算路径的代价估值, 将该路径插入 OPEN 表。扩展到达目标状态或局部空间的边界时, 则求得了一条局部最优路径。由于多目标路径规划问题的所有非支配解路径均为最优, 因此上述过程在求得一条解路径之后, 搜索过程并不终止, 而是继续求解其余局部最优路径。将新生成的路径 p_n 插入 OPEN 表之前进行过滤。若 p_n 的代价估值不被局部空间中的任意一条解路径支配, 则将其插入 OPEN 表。每当求出一条局部非支配解路径, 则对 OPEN 表中所有被支配的路径进行过滤。两次过滤过程保证了每次循环从 OPEN 表中选出将要扩展的路径是不被任意局部解路径支配的。

搜索过程中的启发信息用于从 OPEN 表中选择将要扩展的路径以及对新生成的路径在插入 OPEN 表前进行过滤。路径选择时, OPEN 表中的非支配 OPEN 路径被认为更有希望成为局部非支配路径, 每当发现一条新的局部最优路径时, 将对 OPEN 表中所有被支配的路径进行过滤。扩展路径时, 将新生成的路径 p_n 插入 OPEN 表之前进行过滤, 若 p_n 的代价估值不被局部空间中的任意一条解路径支配, 则将其插入 OPEN 表; 每当求出一条局部非支配解路径时, 对 OPEN 表中所有被支配的路径进行过滤。可见, 搜索过程中利用启发信息能够时刻检查路径之间的支配关系, 对于没有希望的路径提早做出判断, 减少搜索过程的计算量。启发信息越准确, 搜索空间的压缩比例越高。

3.3 学习过程

局部搜索过程结束后, 对于求得的每条局部最优路径, 执行在线学习过程, 对局部空间中各个状态的启发信息进行快速更新。在线学习过程能够使求解过程具有学习能力, 提供更准确的启发信息, 增强算法的引导能力, 过程描述如下:

LEARNING(S_w , CLOSED)

1) 令 $Que = \langle S_w \rangle$

2) 反复执行步骤 a) 一步 b), 直到 Que 为空

a) $S_x = Que.pop()$;

b) 对 S_x 的所有前驱状态 S_y , 若 $S_y \in CLOSED$, 则

i) 对每一个目标函数上的启发信息, 令 $h_{yi} =$

$\min_{S_z \in succ(S_y)} (c_i(y, z) + h_{zi})$, 其中 $1 \leq i \leq N$, N 为目标函数的个数;

ii) $Que.push(S_y)$ 。

局部搜索过程中用集合 CLOSED 记录所有扩展过的状态, 算法调用 LEARNING(S_w , CLOSED) 修改 CLOSED 中所有状态的启发信息。对于状态 S_y , 将每一个目标函数上的启发信息修改为其所有后继状态 S_z 的启发信息中 $c_i(y, z) + h_{zi}$ 的最小值。由于单调的启发函数满足 $h_i \leq c(i, j) + h_j$, 沿局部最优路径修改所有目标函数上的启发函数值, 使得 CLOSED 中的各个状态包含更加充分的启发信息, 增强算法的启发能力, 提高下一次相同目标搜索问题的求解效率; 对每一条局部最优路径执行一次学习过程, 能够保证各个状态的启发函数在每一个目标函数上均为到达目标状态的最优代价估计, 提高了启发信息的准确性。

3.4 移动过程

算法的移动过程沿所求局部最优路径移动到局部空间的一个边界状态, 或探测到移动路径上与假设不一致的环境信息, 移动过程结束, 立即执行下一次局部搜索过程。

MOVE(S_{cur}, p_u)

1) 反复执行步骤 a) 一步 b), 直到 $S_{cur} = S_u$

a) 对 S_{cur} 周围环境信息进行探测, 对于 $(S_{cur}, S_v) \in p_u$, 若 S_{cur} 与 S_v 之间探测到障碍物, 则返回 S_{cur} ;

b) 将 S_{cur} 更新为 S_v 。

移动过程 MOVE(S_{cur}, p_u) 沿局部搜索过程求得的最优路径 p_u 向前移动, 移动过程中对环境信息进行探测, 以便处理存在动态障碍物的环境。若移动路径被障碍物阻塞, 则返回当前所处状态 S_{cur} 并从状态 S_{cur} 执行下一次局部搜索过程, 重新计算新的局部移动路径。这也是实时处理方法的另一个优点。

4 算法最优性与复杂度分析

下面证明局部搜索过程所求路径的局部最优性。

引理 1 若每一个目标函数上启发函数均满足单调限制, 则求解多目标问题的非支配解路径时所扩展节点的代价估值是单调的。

证明: 启发函数的单调性使得对任意的状态 S_u 及其后继状态 S_v , 每一个目标函数上的启发信息 $h_{ui} \leq c_i(u, v) + h_{vi}$, $1 \leq i \leq N$, 因此 $h_u < c(u, v) + h_v$, $g_u + h_u < g_u + c(u, v) + h_v$, 即 $f_u < f_v$ 。

定理 1 局部搜索过程能够求得局部空间内的所有最优路径。

证明: 令 p_n 是深度为 d 的一条路径, 假设 p_n 是被另一条深度为 d 的路径 p_w 支配的, 即 $f_w < f_n$, 此时 p_w 一定尚未被扩展出, 否则 p_n 将在过滤过程从 OPEN 表中删除。由引理 1, 对于 p_w 的任意一条子路径 p_{wi} 满足 $f_{wi} < f_w$, 且存在一条子路径 p_{wi} 与 p_n 同时存在于 OPEN 表中, 因此 $f_{wi} < f_w < f_n$, 路径 p_n 是被支配的, 不会被扩展。由此证明任何被支配的局部路径不会被扩展。

令 p_d 是一条深度为 d 的局部非支配路径, 由引理 1 可知, 对于 p_d 的任意子路径 p_{di} , 若 $f_{di} < f_d$, 则 p_{di} 也是不被任何其他深度为 d 的路径代价估值所支配的。 p_{di} 位于 OPEN 表中时, 若 p_{di} 为一条非支配 OPEN 路径, 则 p_{di} 将被扩展。若 p_{di} 在被 OPEN 表中其他深度小于 d 的路径支配, 则所有支配 p_{di} 的路径因扩展或过滤从 OPEN 表中删除时, p_{di} 将被扩展,

因此局部非支配路径 p_d 的任何子路径都会被扩展,直到 p_d 被找出。由此证明算法能够求出所有局部最优路径集合。

实时多目标启发式搜索算法 RTMHS 的主要空间开销为局部搜索过程 OPEN 表对局部空间进行搜索时存储到达各个状态节点的路径。文献[11]证明了对于元素个数为 L 的 q 维向量集合,其中非支配的向量的平均个数为 $O((\ln L)^{q-1})$ 。算法每一次局部搜索过程都不受问题规模限制,当局部空间最多包含 D 个状态,到达一个状态节点的所有路径中非支配路径个数为 M 时,需要存储的路径个数至多为 MD ,因此 q 个目标函数的实时多目标启发式搜索的最坏空间复杂度为 $O((\ln MD)^{q-1})$ 。另外,算法利用启发信息引导局部搜索过程,能够进一步压缩搜索空间,提高求解的时间效率。

5 实验测试与结果分析

本节采用二维 Grid World 标准测试问题^[3]进行实验测试。实验平台如下:操作系统 Fedora 8.0, CPU Intel Pentium 4 2.80GHz, 内存 1GB DDR。4 个方向的 Grid World 问题初始位置为左上角的格子,目标位置为右下角的格子;向量的每一维元素值在 $[1, 10]$ 之间的整数中随机生成;坐标为 (x_m, y_m) 的状态 S_m 的启发信息 $h(m) = (|x_{S_{goal}} - x_m|, |y_{S_{goal}} - y_m|)$ 。

第一部分实验以两个目标函数的路径规划问题为例,测试局部空间大小不同时对求解时间与路径代价的影响,以及 RTMHS 算法与传统的多目标启发式搜索算法 NAMOA*^[3] 实验结果的比较。随机阻塞 Grid World 状态空间的一些格子。表 1 为不同规模的 Grid World 问题下不同局部搜索深度的测试结果。

表 1 不同规模 Grid World 问题的测试结果

grid_size	Deep	Search time(s)	Path expanded	path_cost
50 * 50	1	0.00347	98	(466,442)
50 * 50	5	0.02217	478	(376,372)
50 * 50	10	0.17184	2043	(403,371)
50 * 50	12	0.62159	3222	(385,323)
50 * 50	15	4.43008	7697	(369,361)
50 * 50	NAMOA *	1.23664	2499	(349,409)
100 * 100	1	0.00951	198	(823,819)
100 * 100	5	0.01076	964	(744,782)
100 * 100	10	0.13425	3954	(716,725)
100 * 100	15	2.23856	12324	(664,836)
100 * 100	20	26.90789	26657	(667,841)
100 * 100	NAMOA *	6.16182	9999	(652,710)
200 * 200	1	0.02153	398	(1804,1770)
200 * 200	5	0.08935	2547	(1632,1546)
200 * 200	10	0.81454	9642	(1585,1511)
200 * 200	15	7.12182	29894	(1599,1420)
200 * 200	20	102.78178	70616	(1634,1284)
200 * 200	NAMOA *	101.53721	39484	(1490,1563)
300 * 300	1	0.05242	598	(2445,2697)
300 * 300	5	0.07298	2846	(2397,2294)
300 * 300	10	1.54822	13939	(2217,2243)
300 * 300	15	11.40734	44177	(2068,2337)
300 * 300	18	44.26377	70519	(2258,2246)
300 * 300	20	416.67279	113868	(2220,2143)
300 * 300	NAMOA *	739.58704	587036	(1897,2368)
400 * 400	1	0.05704	798	(3595,3446)
400 * 400	5	0.08761	3630	(3250,3306)
400 * 400	10	1.63070	17625	(3076,2905)

400 * 400	15	13.40065	56011	(2983,2904)
400 * 400	18	77.57133	97891	(3058,3063)
400 * 400	20	425.16446	141132	(3116,2840)
400 * 400	NAMOA *	2354.42017	158387	(2941,2943)
500 * 500	1	0.09115	998	(4475,4496)
500 * 500	5	0.11374	4640	(3818,3945)
500 * 500	10	2.18166	22983	(3721,3836)
500 * 500	15	40.50118	67068	(3807,3768)
500 * 500	18	227.47186	126875	(3527,3712)
500 * 500	20	537.45630	161562	(3711,3712)
500 * 500	NAMOA *	3827.24829	247819	(3465,3967)

表 1 实验结果表明, NAMOA* 算法直接搜索整个状态空间。由于 OPEN 表需要记录大量的路径信息,导致选择路径扩展时查找速度较慢,求解效率较低。然而,由于对整个状态空间进行彻底搜索, NAMOA* 算法直接计算到达目标状态的路径,能够保证最终移动路径的最优性;选取不同的局部搜索深度时呈现不同的求解能力。深度为 1 时,求解时间最快,路径扩展数最小,但求得的移动路径代价较高,路径质量较差;增大局部搜索深度时,求解时间变长,路径扩展数增加,但能够得到质量较好的移动路径。

第二部分实验进一步对目标函数的个数多于两个的路径规划问题进行测试。考虑求解效率与路径质量的折中,局部搜索深度取 8 至 15。表 2 所列为不同目标函数下分别采用 RTMHS 算法与 NAMOA* 算法的平均求解时间、路径扩展数与移动路径代价。实验结果表明,对于各种不同目标函数情况下的路径规划问题,使用 RTMHS 算法能够得到质量较好的移动路径,求解能力对问题规模的增加不敏感,求解效率也有不同程度的提高。

表 2 目标函数个数不同情况下的测试结果

	RTMHS		NAMOA *	
	Search time(s)	Path expanded	Search time(s)	Path expanded
3-obj				
50 * 50	0.23375	1533	2.36569	2499
100 * 100	1.63267	8878	6.86186	9992
200 * 200	13.16735	18304	91.53367	39632
300 * 300	26.07319	25449	2097.89575	88329
400 * 400	33.47290	33951	2252.72412	158555
500 * 500	33.32592	42955	7691.29736	249505
4-obj				
50 * 50	0.62516	1661	2.73683	2369
100 * 100	5.12892	6056	38.91685	9272
200 * 200	11.70502	12326	408.17169	39836
300 * 300	59.67202	31850	2602.03613	88791
400 * 400	110.32384	43281	4899.79150	158560
500 * 500	116.17815	50143	8965.34082	248759
5-obj				
50 * 50	0.44308	1667	2.33494	2465
100 * 100	5.45776	6626	29.92258	9914
200 * 200	37.53825	22181	276.12829	39993
300 * 300	66.09161	33469	1510.84961	87995
400 * 400	83.67821	43368	4367.62988	159350
500 * 500	106.03901	53787	6243.72461	245176

实验结果表明,动态多目标路径规划方法的优点是:1) 将搜索范围限制在局部空间内,能够避免大量的计算,求解局部最优路径的工作量较小;2) 局部空间的大小独立于整个状态空间,局部搜索过程的执行时间不受问题规模限制,求解效率较高。因此,在求解时间严格受限的情况下,采用实时的启

(下转第 269 页)

建图像的 PSNR 值,用于评估本文算法的性能。图 4 中比较了本文算法、JPEG2000 有损变换算法和 JPEG2000 无损变换算法的 PSNR 值曲线。从图中可以明显地看出,在非常广的压缩范围内,本文算法优于 JPEG2000 有损变换算法和 JPEG2000 无损变换算法 1~2db,也就是说本文算法能更多地提高重建图像的客观质量。在表 1、表 2 中,比较了本文算法、JPEG2000 有损算法和 JPEG2000 无损算法重建图片的 PSNR 值,定量地说明了本文提出算法的优越性。

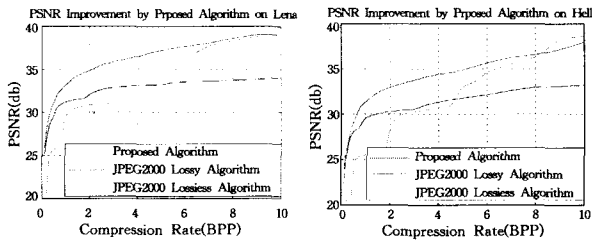


图 4 本文算法与 JPEG2000 有损算法和 JPEG2000 无损算法重建图像客观质量比较

由于客观质量评价标准并不能完全准确地反映主观质量,我们又采用了加权峰值信噪比(WPSNR)作为视觉质量评价标准并画出了包含 CSF 滤波和不包含 CSF 滤波的算法的 WPSNR 值的对比曲线(见图 5)。从图中可以看,视觉质量的提高非常明显。

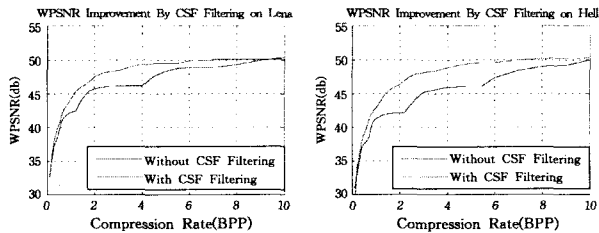


图 5 CSF 滤波对重建图像客观质量的提高

最后,在图 6 中比较了本文算法和 JPEG2000 有损变换算法,从图中可以发现,本文算法明显优于 JPEG2000 有损变换算法。

结束语 本文研究了彩色图像压缩中的预处理算法并提出了一种新的预处理算法。本文算法包含 3 个部分,其一是

RCT 和 CDF 9/7 小波变换的组合,其二是颜色分量加权,其三是颜色对比度敏感函数滤波。实验结果表明,相比 JPEG2000 有损变换算法和 JPEG2000 无损变换算法,本文方法极大地提高了基于小波变换的彩色图像压缩算法的客观质量和视觉质量。



图 6 本文算法与 JPEG2000 有损算法重建图像的比较,左图为本文算法,右图为 JPEG2000 有损算法(0.2BPP)

今后我们将继续研究如何将人类视觉系统(HVS)的其他特性,如掩盖效应等嵌入到彩色图像压缩中去,也将继续研究从变换的角度提高压缩重建图像的质量。

参考文献

- [1] Charrier M, Cruz D S, Larsson M. JPEG2000, the next millennium compression standard for still images[C]// Proc. of the IEEE International Conference on Multimedia Computing and Systems (ICMCS). 1999;131-132
- [2] Taubaman D S, Marcellin M W. JPEG2000 image compression fundamentals, standards and practice, Chinese Edition[Z]. 2004
- [3] ISO/IEC. JPEG 2000—lossless and lossy compression of continuous-tone and bi-level still images; part I minimum decoder[M]. ISO, ISO/IEC, 1999
- [4] Nadenau M J. Integration of human color vision models into high quality image compression[D]. 2000
- [5] Adams M D. Reversible integer-to-integer wavelet transforms for image compression; performance evaluation and analysis[J]. IEEE Transactions on Image Processing, 2000, 9(6): 1010-1024
- [6] Said A, Pearlman W A. A new, fast, and efficient image codec based on set partitioning in hierarchical trees[J]. IEEE Transaction on Circuits and System for Video Technology, 1996, 6(3): 243-250
- [7] Refanidis I, Vlahavas I. Multiobjective heuristic state-space planning [J]. Artificial Intelligence, 2003, 145(1/2): 1-32
- [8] Korf R. Real-time heuristic search [J]. Artificial Intelligence, 1990, 42(2/3): 189-211
- [9] Bulitko V, Lee G. Learning in real-time search; a unifying framework [J]. Journal of Artificial Intelligence Research, 2006, 25: 119-157
- [10] Koenig S, Likhachev M. Real-time adaptive A* [C]// Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems(AAMAS), 2006; 281-288
- [11] Rayner D C, Davison K, Bulitko V, et al. Real-time heuristic search with a priority queue [C]// Proceedings of the 20th International Joint Conference on Artificial Intelligence(IJCAI). 2007; 2372-2377
- [12] Bentley J L, Kung H T, Schkolnick M, et al. On the average number of maxima in a set of vectors and applications [J]. Journal of the ACM, 1978, 25(4): 536-543
- [13] Stewart B S, White C C. Multiobjective A* [J]. Journal of the ACM, 1991, 38(4): 775-814
- [14] Dasgupta P, Chakrabarti P P, De Sakar S C. Multiobjective heuristic search [M]. Vieweg: Braunschweig/Wiebaden, 1999
- [15] Mandow L, De la Cruz J L P. Multicriteria heuristic search [J]. European Journal of Operational Research, 2003, 150(2): 253-280
- [16] Harikumar S, Kumar S. Iterative deepening multiobjective A*

参考文献

(上接第 239 页)

发式搜索在线地对问题进行求解,通过选取合理的局部空间能够在保证路径质量的同时减小工作量,避免过长的停留时间,快速地到达目标状态,很好地解决一系列多目标路径规划问题。

结束语 通过在线交替执行局部搜索过程、学习过程与移动过程,实时的路径规划方法在预先假设的环境下试探性地进行求解,移动的同时探测周围环境,尽早发现不一致的环境信息,尤其当环境中存在大量不确定的信息时,能够避免大量的重规划过程,从而提高问题求解的效率。