

文章编号:1001-9081(2010)11-2998-04

基于侧影轮廓的三维模型快速重建

阮孟贵^{1,2}, 章毓晋^{1,2}

(1. 清华大学 信息科学与技术国家实验室, 北京 100084; 2. 清华大学 电子工程系, 北京 100084)

(rmg07@mails.tsinghua.edu.cn)

摘要:提出一种基于侧影轮廓进行三维模型重建的新方法,将传统的三维锥形交叉问题转换成二维轮廓交叉问题。首先,将不同视角下的二维侧影轮廓反投影到若干个平行的三维平面上,然后在三维平面上计算所有反投影轮廓的交叉轮廓,最后对相邻三维平面上的交叉轮廓进行匹配并重建物体的表面。理论分析和实验结果表明该算法的时间复杂度和视角数目呈线性关系。由于该方法主要以增加视角数目来提高模型的精确度,所以比三维锥形交叉的方法能更快速地重建物体精确的三维模型。

关键词:三维模型重建;可见外壳;交叉轮廓

中图分类号: TP391.41 **文献标志码:** A

Fast 3D model generation from silhouettes

NGUYEN Manh-quy^{1,2}, ZHANG Yu-jin^{1,2}

(1. National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China;

2. Department of Electronic Engineering, Tsinghua University, Beijing 100084, China)

Abstract: In this paper, a new approach for fast 3D model generation from silhouettes was proposed, in which the traditional 3D cones intersection problem was converted to the 2D silhouettes intersection problem. Firstly, the 2D silhouettes of different viewpoints were projected back to parallel 3D planes, then the intersectional silhouettes of all back-projections on the 3D plane were calculated, finally the corresponding points between the intersectional silhouettes of two neighboring 3D planes were matched, so the mesh of 3D model was directly obtained. Both theoretical analysis and experimental results show that the time complexity of the proposed algorithm is increased linearly with the number of viewpoints. Since the proposed method mainly improves the accuracy of 3D model by increasing the number of viewpoints, it makes out the precise generation of 3D models faster than the 3D cones intersection method.

Key words: 3D model generation; visual hull; intersectional silhouette

0 引言

近年来,三维模型已经广泛应用于三维电脑游戏、三维动画电影、医疗诊断辅助系统、遗产文化保护等方面。三维模型重建的方法可分为主动方法和被动方法两大类^[1]。主动方法主要以仪器直接去测量物体表面的三维信息,获得三维点云,再用点云算法重建物体的表面以及获得三维模型,主动方法可以提供较高精度的三维模型,但需要非常昂贵的设备。被动的方法主要研究如何根据相机在不同视角拍摄的图像序列进行物体的三维模型重建。基于图像的三维模型重建运算量较大(运算速度较慢),并且易受到噪声、光照变化、阴影等因素的影响。所以一般要从图像提取一些鲁棒的特征,然后进行匹配以获得三维模型。

基于侧影轮廓的三维重建方法也称为可见外壳(Visual Hull)生成方法,最早由 Laurentini 提出^[2],是被动三维模型重建方法的一种。这种方法主要利用物体在图像中的侧影轮廓线来生成三维模型(见图1)。基于侧影轮廓的三维模型重建方法具有较强的鲁棒性,能够较好地保证几何和拓扑结构的一致性,并且可以通过增加视点数目来得到越来越精确的三维模型。

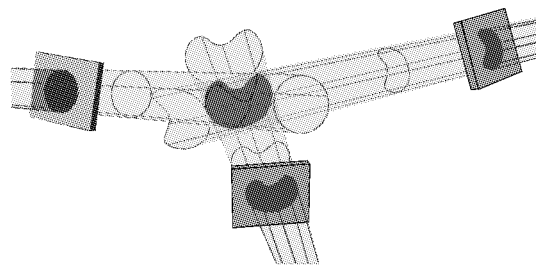


图1 从不同视角的侧影轮廓恢复3D模型的示意图

近年来有较多学者提出了不同的生成可见外壳的算法^[3-7],主要思路是将侧影轮廓反投影到三维空间(用三维锥形表示),然后计算若干三维锥形的交叉以得到三维模型。例如:Matusik用若干个三维面来表示三维锥形(见图2(a)),并通过计算不同视角下的轮廓相交来得到三维多边形片^[3];而Franco用若干个三维线段来表示三维锥形(见图2(b)),并通过计算不同视角下的轮廓相交来得到三维线段^[4-5]。Franco还对三维锥形交叉算法进行分析,指出这类算法时间复杂度为 $O(n^2 m^2 q \log(mq))$,其中 n 为视角的数目, m 为场景中物体的个数, q 为物体侧影轮廓的最大点数^[5]。从Franco的分析可见,基于三维锥形交叉重建方法的算法时间复杂度与视角数目成平方关系。所以,该方法在需要逼真和精确的

收稿日期:2010-05-21;修回日期:2010-06-24。 基金项目:国家自然科学基金资助项目(60872084)。

作者简介:阮孟贵(1983-),男,越南河内人,硕士研究生,主要研究方向:三维重建、目标分割; 章毓晋(1954-),男,江苏江阴人,教授,博士生导师,主要研究方向:图像工程。

三维模型时受到计算时间的约束。

本文提出一种基于侧影轮廓的三维模型重建新方法,本文算法的时间复杂度为 $O(knmq \log(mq))$, k 为三维平面的个数(一般为常数), n 为视角的数目, m 为场景中物体的个数, q 为物体侧影轮廓的最大点数。可见,本文方法的算法时间复杂度与视角数目呈线性关系,更适合快速地重建精确的三维模型。本文方法主要分为两个步骤:首先要计算三维平面上的交叉轮廓以及获得物体表面的点集,然后对相邻的两个三维平面的交叉轮廓进行匹配以获得物体的表面。

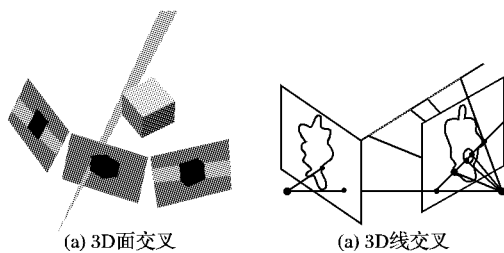


图2 3D 锥形交叉实现方法

1 物体表面点集的生成

1.1 反投影方程

设 P_i 为相机 i ($i = 1, 2, \dots, n$) 的参数矩阵,则投影方程为:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = P_i \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 & p_7 \\ p_8 & p_9 & p_{10} & p_{11} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

$$\Rightarrow \begin{cases} X_I = \frac{x}{z} = \frac{p_0 X + p_1 Y + p_2 Z + p_3}{p_8 X + p_9 Y + p_{10} Z + p_{11}} \\ Y_I = \frac{y}{z} = \frac{p_4 X + p_5 Y + p_6 Z + p_7}{p_8 X + p_9 Y + p_{10} Z + p_{11}} \end{cases} \quad (2)$$

$$\Leftrightarrow \begin{cases} (X_I p_8 - p_0) X + (X_I p_9 - p_1) Y = Z p_2 + p_3 - X_I (Z p_{10} + p_{11}) \\ (Y_I p_8 - p_4) X + (Y_I p_9 - p_5) Y = Z p_6 + p_7 - Y_I (Z p_{10} + p_{11}) \end{cases} \quad (3)$$

其中: (X, Y, Z) 为空间坐标点, (X_I, Y_I) 为图像坐标点。

如果将给定三维平面方程 $R: aX + bY + cZ + d = 0$ 与方程组(3)进行联立求解,可得到相机 i 中的轮廓 S_i 反投影到三维平面 $R: aX + bY + cZ + d = 0$ 的反投影轮廓 S_i^R 。

1.2 二维轮廓交叉

设 S_i^R 是相机 i 中的轮廓 S_i 在三维平面 $R: aX + bY + cZ + d = 0$ 上的反投影轮廓 ($i = 1, 2, \dots, n$)。那么交叉轮廓 $S^R = \bigcap_{i=1, \dots, n} S_i^R$ 是物体表面和三维平面 $R: aX + bY + cZ + d = 0$ 的三维交叉轮廓,是物体表面在三维平面 R 上的点集。这步计算是本文算法最关键的步骤,因为它占了整个算法的大部分时间。

关于轮廓交叉问题,有较多学者提出了不同的算法。当轮廓的点数较大时,轮廓点用浮点数表示且性能较好的算法主要有 Vatti 算法^[8]、Greiner 算法^[9]和 Martinez 算法^[10]。Vatti 根据轮廓的局部最大和最小点将轮廓分段,并定义交叉规则,计算出轮廓之间的交叉点,并组成交叉轮廓。该算法由 Murta 实现^[11],经过多次的改进算法速度较快并且性能比较稳定。Greiner 采用链表的数据结构,利于添加交叉点,将交叉点分成入口和出口的交叉点,然后根据这些入口和出口交

叉点来组成交叉轮廓。Greiner 算法的时间复杂度为 $O(q^2)$, 其中 q 为轮廓点数。最近, Martinez 等人^[10]采用平面扫描方法^[12],并用一个边的词典来构成交叉轮廓。Martinez 算法的时间复杂度为 $O((q + l) \log q)$, 其中 q 为轮廓点数, l 为轮廓交叉点数。Martinez 算法在计算交叉点的过程中动态地将新的交叉点插入优先级队列,但在实际中轮廓边与其他轮廓边交叉两次以上的概率非常小,所以这样的处理方式使得当边的总数 q 和交叉点数 l 较大时的算法性能下降,即运算速度慢。

本文提出的轮廓交叉算法也采用平面扫描方法^[12],但使用了新的轮廓数据结构。首先,采用平面扫描方法计算出轮廓之间所有的交叉点,然后计算出属于交叉轮廓的线段,最后将这些线段连接成交叉轮廓。为了提高算法的效率,本文提出一种独特的轮廓数据结构,即数组和双链表结合的数据结构。如图3所示,轮廓的原有节点 V_i ($i = 1, 2, \dots, n$) 以数组方式存储,而交叉节点(例如: I_m 和 I_p) 以双链表方式存储。采用本结构使得在平面扫描过程中,以 $O(1)$ 时间去访问轮廓边,同时也以 $O(1)$ 时间去存储交叉节点。而 Martinez 算法是以 $O(\log q)$ 时间去存储交叉节点。本文算法的时间复杂度为 $O(q \log q)$, 其中 q 为轮廓点数。

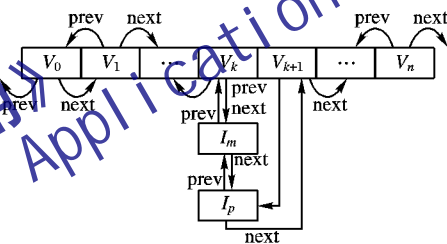


图3 轮廓的数组和双链表结合的数据结构

图4给出本文的轮廓交叉算法与 Vatti 算法、Greiner 算法和 Martinez 算法的时间比较。可见本文算法性能优于这些对比的算法,本文算法的速度相对于 Martinez 算法提高了两倍,轮廓点数增大时本文算法性能仍然很稳定,没有出现算法性能快速下降的情况。

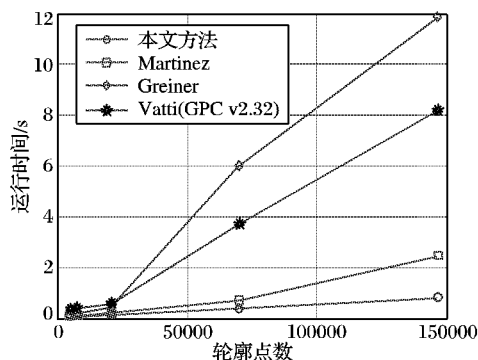


图4 轮廓交叉算法的性能比较

2 物体表面生成

上述的过程只生成物体表面上的点集,还需要连接两两相邻三维平面上的轮廓点,并添加三角片以重构物体表面。关于两个相邻平面间的轮廓点匹配算法,最早 Fuchs 等人^[13]采用图结构表示方法,并进行优化找出两层平面间的轮廓点对应关系。之后, Meyers 等人^[14]将平面之间物体表面重构问题分成4个子问题:1)多个与多个轮廓对应问题;2)单个与单个轮廓的点对应问题(一般采用最小面积准则);3)分支问题;4)曲面重建问题。Cong 等人^[15]采用相等重要准则(equal

importance criterion) 优化曲面的光滑性, 定义了“曲面区域”, 是将两个轮廓进行异或运算的区域。Cong 算法需要解偏微分方程来重构物体的表面, 算法比较复杂。

本文提出的匹配算法是利用了交叉轮廓的特性来提高匹配的精确度, 并能够保证拓扑结构。根据 1.2 节的介绍, 三维平面上的交叉轮廓是由不同视角的反投影轮廓线段组成的, 如图 5(a) 和 (c) 所示, 不同颜色的线段表示不同反投影轮廓的线段, 可见两个相邻三维平面上的交叉轮廓的线段构造非常相似。这样, 平面间的交叉轮廓匹配问题被转换成线段匹配问题, 并且能更好保证拓扑结构。对于多个与多个轮廓对应问题, 本文算法采用平面扫描方法^[12], 找出轮廓间的对应关系。对于多个轮廓对应一个轮廓的问题(图 5(c)), 可以利用交叉轮廓的线段信息将这些轮廓并接成一个轮廓, 然后进行匹配(见图 5(d))。

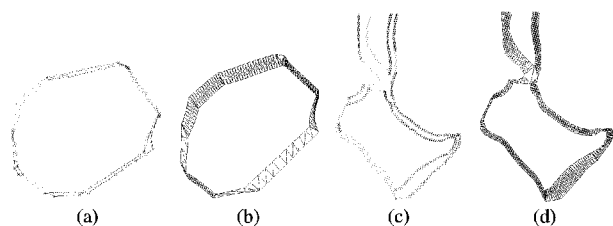


图 5 本文三维平面间的交叉轮廓匹配方法

对于线段点的匹配问题本文采用基于 Delaunay 准则的匹配算法^[16], 设上层平面的线段为 q_0, q_1, \dots, q_m ; 下层平面的线段为 p_0, p_1, \dots, p_n ; 初始时 $q_0 p_0$ 为工作边。设当前的工作边为 $q_i p_j$, 比较两个三角片 $q_i p_j p_{j+1}$ 和 $q_i p_j q_{i+1}$, 并选择最小内角为最大的三角片, 设置新的工作边, 重复上述过程。

图 5(b)、(d) 为图 5(a)、(c) 的对应交叉轮廓匹配结果, 实验表明本文算法能够非常有效地得到物体表面, 并能够保证拓扑结构。本文提出的两个相邻平面间的曲面重构算法的时间复杂度均为 $O(q \log q)$, 其中 q 为轮廓点数。

3 本文 3D 模型重建算法复杂度分析

设 k 为 3D 平面的个数(一般为常数), n 为视角的数目, m 为场景中物体的个数, q 为物体侧影轮廓的最大点数。本文算法分为两个步骤: 1) 生成物体表面点集; 2) 对两个相邻三维平面的交叉轮廓进行匹配重建物体的三维模型。

首先, 要估算物体表面点集生成的复杂度, 这个步骤需要计算在 k 个平面上的 n 个反投影轮廓的交叉轮廓, 算法时间复杂度为 $O(knmq \log(mq))$ 。

其次, 生成两个相邻三维平面之间的物体表面的时间复杂度为 $O(mq \log(mq))$, 所以生成整个物体表面的时间复杂度为 $O(kmq \log(mq))$ 。

综合考虑两个步骤, 算法的总时间复杂度为 $O(knmq \log(mq))$, 可见本文算法的时间复杂度和视角数目 n 呈线性关系。前面已指出, 基于三维锥形交叉的可见外壳算法的复杂度为 $O(n^2 m^2 q \log(mq))$, 即和视角数目 n 成平方关系。由于可见外壳生成方法主要以增加视角的数目来得到更为精确的三维模型, 所以视角的数目越多时, 本文算法的优越性就会越大。

4 实验结果和讨论

实验的环境为 PC P4 3.0 GHz, 512 MB 内存, Windows XP Home, 采用两组图像序列 Alien 和 Predator^[5] 进行了测试

(结果示例见图 6), 表 1 给出测试图像序列的参数。

表 1 测试序列参数

测试序列	图像个数	图像大小	轮廓平均点数
Alien	24	1900 × 1600	9387
Predator	24	1950 × 1750	10516

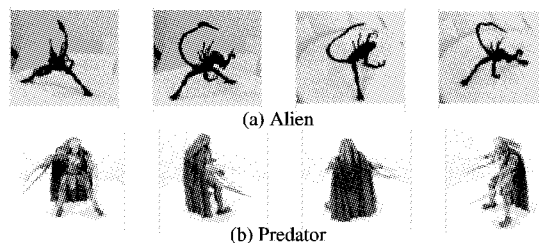


图 6 两组测试序列图像(部分)

图 7~8 给出对应本文方法重建出的 Alien 和 Predator 三维模型, 实验结果表明本文方法能够精确地重建物体的三维模型。



图 7 Alien 的三维模型重建结果

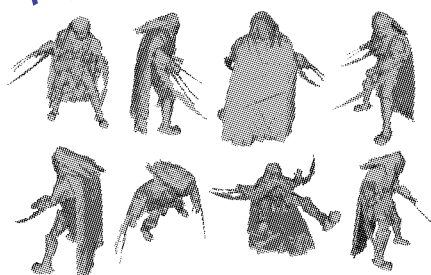


图 8 Predator 的三维模型重建结果

表 2 为本文方法与 Franco 的方法 (EPVH 1.0 和 EPVH 1.1) 和 Lazebnik 的方法比较的结果。因为 Franco 和 Lazebnik 只给出视角数目为 24 时的重建结果, 所以本文仅对此情况进行了比较。对于 Alien 测试序列图 EPVH 1.1 算法运行速度最快, 而对 Predator 测试序列图本文算法运行最快。因为本文算法和视角数目呈线性关系, 所以在视角数目增大时, 本文提出的算法比现有的算法更快。图 9 给出对本文算法的视角数目与运行时间进行实际测试而得到的关系曲线, 实验结果表明本文提出算法的实际性能比线性还好。

表 2 可见外壳算法性能比较

测试序列	方法	顶点个数	面个数	运行时间/s	CPU 主频/GHz
Alien	本文方法	189 194	379 079	167	3.0
	EPVH 1.0	171 752	343 500	3 363	3.4
	EPVH 1.1	171 752	343 500	119	3.4
	Lazebnik	209 885	419 770	532	3.4
Predator	本文方法	313 124	647 771	112	3.0
	EPVH 1.0	306 152	612 420	5078	3.4
	EPVH 1.1	306 152	612 420	136	3.4
	Lazebnik	375 345	750 806	737	3.4

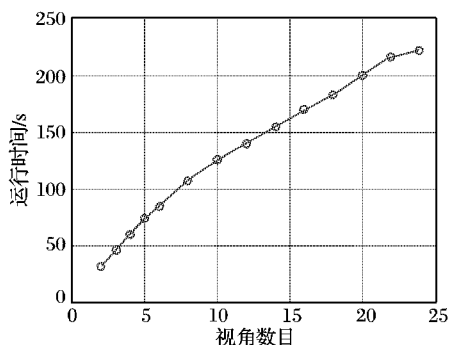


图 9 本文可见外壳算法性能

5 结语

理论分析和实验结果表明,本文提出的快速三维模型重建算法和视角数目呈线性关系。对于以增加视角数目来得到更精确的三维模型的可见外壳生成方法而言,本文算法能比文献[4-6]更快速地重建三维模型。另外,本文算法可以通过增加平行三维平面的个数获得更密集的三维模型重建效果,对于密集高精度三维模型的重建提供了很好的基础。

本文提出的方法易于硬件移植,实现三维模型重建实时系统。借助对场景的实时重建,可以方便地实现目标跟踪和动作识别等。近年来,较多的二维识别问题正逐渐向三维扩展,如三维人脸识别、基于三维重建的故障识别等,可见三维模型重建有非常好的应用前景。

参考文献:

- [1] 章毓晋. 图像工程——图像理解: 下册[M]. 2版. 北京: 清华大学出版社, 2007: 60-72.
- [2] LAURENTINI A. The visual hull concept for silhouette-based image understanding [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994, 16(2): 150-162.
- [3] MATUSIK W, BUEHLER C, RASKAR R, et al. Image-based visual hulls [EB/OL]. [2009-12-12]. <http://people.csail.mit.edu/wojciech/IBVH/ibvh.pdf>

- [4] FRANCO J-S, BOYER E. Exact polyhedral visual hulls [C]// Proceedings of the Fourteenth British Machine Vision Conference. London: BMVA Press, 2003: 329-338.
- [5] FRANCO J-S. Efficient polyhedral modeling from silhouettes [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(3): 414-427.
- [6] LAZEBNIK S, FURUKAWA Y, PONCE J. Projective visual hulls [J]. International Journal of Computer Vision, 2007, 74(2): 137-165.
- [7] KIECK H, HEIDRICH W, VOGELGSANG C. Shape from contours and multiple stereo — A hierarchical, mesh-based approach [C]// Proceedings of the First Canadian Conference on Computer and Robot Vision. Washington, DC: IEEE, 2004: 76-83.
- [8] VATTI B R. A generic solution to polygon clipping [J]. Communications of the ACM, 1992, 35(7): 56-63.
- [9] GREINER G, HORMANN K. Efficient clipping of arbitrary polygons [J]. ACM Transactions on Graphics, 1998, 17(2): 71-83.
- [10] MARTINEZ F, RUEDA A J, FEITO F R. A new algorithm for computing Boolean operations on polygons [J]. Computer & Geosciences, 2009, 35(6): 1177-1185.
- [11] MURTA A. General polygon clipper library [EB/OL]. [2009-12-25]. <http://www.cs.man.ac.uk/~tobczalan/software/>.
- [12] de BERG M, OTFRIED CHEONG, van KREVELD M, et al. Computational geometry: algorithms and applications [M]. Third edition. Berlin: Springer, 2008: 20-29.
- [13] COHEN H, KEMM Z M, USELTON S P. Optimal surface reconstruction from planar contours [J]. Communications of the ACM, 1977, 20(2): 693-702.
- [14] MEYERS D, SKINNER S, SLOAN K. Surfaces from contours [J]. ACM Transactions on Graphics, 1992, 11(3): 228-258.
- [15] CONG G, PARVIN B. An algebraic solution to surface recovery from cross-sectional contours [J]. Graphical Models and Image Processing, 1999, 61(4): 222-243.
- [16] 纪凤欣, 欧宗瑛, 秦绪佳. 基于 Delaunay 三角剖分的层析图像离散数据表面重建算法 [J]. 工程图学学报, 2001, 22(2): 53-58.

(上接第 2883 页)

参考文献:

- [1] DIJKSTRA E. A note on two problems in connexion with graphs [J]. Numerische Mathematik, 1959, 1(1): 267-271.
- [2] FREDMAN M L, TARJAN R E. Fibonacci heaps and their uses in improved network optimization algorithms [J]. Journal of the Association for Computing Machinery, 1987, 34(3): 596-615.
- [3] KÖHLER E, MÖHRING R, SCHILLING H. Fast point-to-point shortest path computation with arc-flags [EB/OL]. [2009-12-12]. <http://www.math.tu-berlin.de/coga/people/schilling/pub/20061113-DIMACS.pdf>.
- [4] LAUTHER U. An extremely fast, exact algorithm for finding shortest paths in static networks with geographical background [EB/OL]. [2009-12-12]. <https://gor.uni-paderborn.de/Members/AC06/LAUTHER.PDF>
- [5] LAUTHER U. Slow preprocessing of graphs for extremely fast shortest path calculations [C]// Lecture at the Workshop on Computational Integer Programming. ZIB: [s.n.], 1997.
- [6] GOLDBERG A V, HARRELSON C. Computing the shortest path: A* search meets graph theory [C]// Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). Vancouver: SIAM, 2005: 156-165.
- [7] GOLDBERG A V, WERNECK R F. Computing point-to-point shortest paths from external memory [C]// Proceedings of the 7th Work-

shop on Algorithm Engineering and Experiment (ALENEX). Vancouver: SIAM, 2005: 26-40.

- [8] GOLDBERG A V, KAPLAN H, WERNECK R F. Reach for A*: Efficient point-to-point shortest path algorithms [C]// ALENEX: proceedings of the 8th Workshop on Algorithm Engineering and Experiment. Vancouver: SIAM, 2006.
- [9] SANDERS P, SCHULTES D. Highway hierarchies hasten exact shortest path queries [C]// Proceedings of the 13th Annual European Symposium. Berlin: Springer, 2005: 568-579.
- [10] SANDERS P, SCHULTERS D. Engineering highway hierarchies [C]// Proceedings of the 14th Annual European Symposium on Algorithms. Berlin: Springer, 2006: 804-816.
- [11] KNOPP S, SANDERS P, SCHULTES D, et al. Fast computation of distance tables using highway hierarchies [R]. Karlsruhe: University of Karlsruhe, Faculty of Informatics, 2006.
- [12] 李清泉, 郑年波, 徐敬海, 等. 一种基于道路网络层次拓扑结构的分层路径规划算法 [J]. 中国图象图形学报, 2007, 7(12): 1280-1285.
- [13] 李锴, 钟耳顺, 曾志明, 等. 基于网络分层拓扑结构的最优路径算法 [J]. 中国图象图形学报, 2006, 11(7): 1004-1009.
- [14] 高松, 陆峰. 一种基于路网等级启发式策略的路径搜索算法 [J]. 地球信息科学学报, 2009, 11(2): 151-156.