引文格式：徐唐进，张安民，高邈，等．动态水深环境下的无人艇路径规划［J］．测绘科学，2021，46(6)：180-189.(XU Tangjin，ZHANG Anmin，GAO Miao，et al. Path planning for unmanned surface vehicle based on dynamic water depth environment［J］. Science of Surveying and Mapping，2021，46(6)：180-189.)DOI：10.16251/j.cnki.1009-2307.2021.06.026.

动态水深环境下的无人艇路径规划

徐唐进[1]，张安民[1,2]，高　邈[1]，刘　帅[3]，栗宝鹃[4]

(1. 天津大学 海洋科学与技术学院，天津　300072；2. 天津市港口环境监测工程技术中心，天津　300072；
3. 天津航海仪器研究所，天津　300131；4. 中水北方勘测设计研究有限责任公司，天津　300222)

摘　要：针对 D* Lite 算法存在路径转折点多、难以进行轨迹跟踪控制、没有顾及动态水深变化等不足，该文引入水深危险度，进行安全路径的选优，以提高所规划路径的安全性，并采用视线检查算法和懒惰更新改进 D* Lite 算法进行路径规划，提出了 LT-D* Lite 算法。该文采用电子海图水深叠加潮汐水位构建动态水深模型，同时建立水深危险度代价函数对最优安全节点进行选择，平衡路径的安全性。算法在拓展节点过程引入视线检查算法，将节点拓展方式从八邻域拓展为任意角度方式并使路径平滑，通过动态水深调整后的浅水区域进行路径调整。仿真实验表明，该文所提出的算法具有规划路径短且路径平滑的特点，同时可引导无人艇避开浅水深区域，显著降低路径危险度。

关键词：电子海图；无人艇；动态环境；路径规划；D* Lite

【中图分类号】P208　　**【文献标志码】**A　　**【文章编号】**1009-2307(2021)06-0180-10

Path planning for unmanned surface vehicle based on dynamic water depth environment

XU Tangjin[1]，*ZHANG Anmin*[1,2]，*GAO Miao*[1]，*LIU Shuai*[3]，*LI Baojuan*[4]

(1. School of Marine Science and Technology，Tianjin University，Tianjin 300072，China；
2. Tianjin Harbor Environment Monitoring Engineering Center，Tianjin 300072，China；
3. Tianjin Navigation Instrument Research Institute，Tianjin 300131，China；
4. China Water Resources Beifang Investigation，Design and Research Co.，Ltd.，Tianjin 300222，China)

Abstract：In view of the shortcomings of the D* Lite algorithm such as many path turning points，difficulty in trajectory tracking control，and failure to take into account environmental factors such as dynamic water depth changes，this paper introduced the risk of water depth and selected the best safe route to improve the safety of the planned route. According to the operational requirements of hydrographic survey in shallow water area，the line of sight checking algorithm and lazy testing were used to improve the D* Lite algorithm for path planning，and the light and depth-D* Lite (LT-D* Lite) algorithm was proposed. Dynamic bathymetric model was constructed by superimposing the tidal water level with the electronic chart bathymetry，and the optimal safety node was selected by establishing the bathymetric risk cost function to balance the safety of the path. A line of sight checking algorithm was introduced in the process of expanding nodes to expand the node expansion mode from eight neighborhoods to any angle mode and make the path smooth，and the path was adjusted through the shallow water area after dynamic bathymetry adjustment. Simulation experiments showed that the algorithm proposed in this paper had the characteristics of short planning path and smooth path，and at the same time could guide the unmanned surface vehicle to avoid the shallow water depth area，which significantly reduced the risk of path.

Keywords：electronic navigation chart；unmanned surface vehicle；dynamic environment；path planning；D* Lite

作者简介：徐唐进(1996—)，男，江苏南通人，硕士研究生，主要研究方向为无人艇路径规划。
E-mail：tjxu4914@tju.edu.cn

通信作者：张安民 教授 E-mail：anmin_zhang@126.com

收稿日期：2021-04-13
基金项目：国家重点研发计划项目(2018YFC1407400)

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

0　引言

随着无人技术的发展，水面无人艇(unmanned surface vehicle，USV)在水质监测、海底勘测、海上巡航和海上运输等方面的应用越来越广[1]。相对于有人船，USV 较多应用于近岸、港口等浅水水域。USV 等航行器在近岸等浅水域中作业时，需对安全水深区和碍航区进行准确划分。文献［2］为简化分析，在障碍物边界预留船舶安全缓冲区，缓冲区半径取 USV 预留安全距离和旋回半径的较大值，以确保船舶安全航行。文献［3］将岛礁等障碍物抽象为规则圆进行 USV 的路径规划。一方面，真实的水下地形复杂，且测区的瞬时水深受到潮汐的影响会不断变化，碍航区也因此发生变化；另一方面，为满足多种作业需求，USV 装载多种探测设备，导致 USV 吃水增大，USV 受风浪流等因素影响产生摇摆和升沉运动会改变 USV 距水底的垂向距离，而带来安全水深的变化。根据上述学者简单的障碍物界定方法难以准确甄别所有的碍航区，很容易导致部分不满足航行条件区域被划入可航行水域，或者由于碍航区扩大化而影响 USV 进行测区全覆盖测量。简单的矢量线路径规划方法难以适应 USV 航行过程中由于潮汐等因素带来的碍航区动态变化，栅格化的环境模型，能较准确地反映环境信息，与路径规划搜索算法相结合，实现路径的动态调整[4]。

路径规划算法分为图搜索规划算法、随机采样方法、曲线插值方法、机器学习方法、仿生智能方法和动态优化方法等[5]。基于网格图的图搜索规划算法属于确定性算法，具有较高效率。A* 算法[6]是网格搜索算法，基于邻域节点进行拓展连接，最终连接节点形成路径，其有效解决在静态环境下的寻找最优路径问题。文献［7］设计 Lazy Theta* 算法，利用视线检查算法实现任意方向规划，克服 A* 算法只能沿着栅格中心移动的局限性，并根据懒惰更新减少拓展节点数，提高算法效率。相对于 A* 算法及其改进算法，D* 算法[8]则实现了动态环境下的路径规划。但是 D* 算法较为复杂，判断条件较多，算法效率低。文献［9］提出 LPA*(lifelong planning A*)算法，引入环境增量参数，在重规划过程中，更新预规划的搜索信息，实现了在定起点定目标点的未知环境下遇见障碍物时的快速重规划。针对上述问题，文献［10］提出 D* Lite 算法，在重规划的过程中，只需考虑当前节点到目标点的路径信息，实现变起点、定目标点的路径规划。

上述算法，在栅格环境下进行规划，受限于八邻域的拓展方式，最终规划的路径不利于进行轨迹跟踪控制。为提高规划算法的搜索效率和规划的路径质量，不断有学者对成熟的启发式算法进行改进应用于动态复杂环境下机器人路径规划，缩短搜索空间，提高路径规划质量。文献［11］提出增强 D* Lite 算法，对路径经过障碍进行检测，进行路径安全改善。文献［12］对 D* Lite 算法引入 Bresenham 算法[13]，减少不必要的转折，但同时需要对画线算法填充的栅格位置进行计算，极大增加规划效率。在国外，文献［14］提出 Field D* 算法，通过对栅格进行线性插值，使路径点不局限于端点，按照任意角度进行规划，生成平滑路径。但是插值过程中线性估计的误差会随路径点代价的更新不断累积，导致插值点位置估计发生偏差，前后路径的连线不共线。

本文在电子海图的静态水深模型的基础上，根据潮汐数据推算实时变化水位，形成动态的栅格化的水深模型，在 D* Lite 算法的代价函数中引入水深危险度代价。同时，本文为解决 D* Lite 算法规划的路径存在路径不平滑和易碰撞障碍物的问题，引入视线检查算法和懒惰更新法，使得规划出的路径安全、平滑且较短。

1　环境建模

根据 USV 航行及探测设备安全所需的最小安全水深，将环境区域划分为安全水深区和碍航区[15]。碍航区的划定包括人工碍航区和浅水碍航区。人工碍航区包括人为划定的特殊区域，此类碍航区的位置和边界数据可在电子海图中直接获取。浅水碍航区是指影响航行器安全航行的浅水深区域，由相关障碍物地理要素(主要包括岸线、岛、礁和码头等数据)和航行海区内低于航行器最小安全水深值的区域组成。随着时间的变化，瞬时水深不断变化，碍航区和安全水深区的范围可能会因此发生相应的变化。通常，USV 在航行中难以实时获取大范围高精度水深信息，需要利用通过海图数据建立的静态水深模型和潮汐模型构建瞬时水深环境模型。

1.1　静态安全水深模型

本文以 S-57 标准矢量电子海图水深数据为基础，将海图坐标转换成墨卡托海图[16]，并利用栅格图来描述空间对象的信息，构建静态安全水深模型。

栅格法将海图划分为大小相等的网格，并且每个栅格单独记录该栅格点的地形信息。通过 x 和 y 描述位置信息，对于判定为安全水深区的栅格存储有静态水深值 h_s。通过含障碍的样条插值算法可以获得非障碍区每个栅格的水深预测值，进而能够以获得的预测水深值为参考计算每个栅格的

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

水深危险度[17]。样条函数的计算方法为

$$D(x,y)=F(x,y)+\sum_{j=1}^{N}\lambda_j R(r_j) \tag{1}$$

式中：$D(x,y)$ 为待估点的水深值。规则样条函数计算公式如式(2)所示[18]。

$$F(x,y)=a_1+a_2x+a_3y \tag{2}$$

$$R(r)=\frac{1}{2\pi}\left\{\frac{r^2}{4}\left[\ln\left(\frac{r}{2\tau}\right)+c-1\right]+\tau^2\left[K_O\left(\frac{r}{\tau}\right)+c+\ln\left(\frac{r}{2\pi}\right)\right]\right\} \tag{3}$$

式中：r 为点与样本点之间的距离；τ^2 为权重系数；K_o 为修正贝塞尔函数；c 为大小等于 0.577 215 的常数；λ_j，a_i 为通过求解线性方程组而获得的系数；r_j 为点 (x,y) 到第 j 个输入点的距离。

在计算的过程中，插值输出的部分按照栅格尺寸被划分为 x 方向和 y 方向上的大小相等的矩形栅格。通过上述方法，建立具有静态水深信息的栅格环境模型。

1.2 潮汐水位数据

USV 在航行过程中，不可避免地受到潮汐等影响导致水深环境发生变化，影响整体的路径规划。公开潮汐数据仅包含高低潮潮时潮位，USV 航行时需要逐时潮位数据，本文选用三次样条插值对潮汐表的潮位数据进行插值处理，得到潮位随时间的变化函数。首先建立潮位随时间连续变化函数 $H(t)$，已有的潮时节点组 (t_n,H_n)，相邻节点组 (t_{n-1},H_{n-1}) 和 (t_n,H_n) 之间通过三次样条函数 $P(t)$ 进行插值求解，其表达式为：

$$P_n=P(t_n)=a_0+a_1t_n+a_2t_n^2+a_3t_n^2 \tag{4}$$

式中：a_0，a_1，a_2，a_3 为待定系数。每段中均存在 4 个待定系数，因此需建立 $4n$ 个方程进行求解，如式(5)所示。

$$\left.\begin{aligned}
&P_{n-1}(t_n)=P_{n-1}(t_n) && (n=1,\cdots,m-1)\\
&P'_{n-1}(t_n)=P'_n(t_n) && (n=1,\cdots,m-1)\\
&P''_{n-1}(t_n)=P''_{n-1}(t_n) && (n=1,\cdots,m-1)\\
&P(t_n)=H_n && (n=0,\cdots,m)\\
&P'_n(t_0)=0\\
&P'_m(t_m)=0
\end{aligned}\right\} \tag{5}$$

求解上述高阶矩阵即可获得潮位随时间变化函数。

1.3 瞬时水深数据

本文中某时刻栅格的瞬时水深可由栅格点的静态水深和潮位进行叠加计算获得。瞬时水深可表示为[19]：

$$S(x,y,t)=D(x,y)+T(x,y,t)+\Delta(x,y,t) \tag{6}$$

式中：$S(x,y,t)$ 为 (x,y) 栅格点在时刻 t 的瞬时水深值；$D(x,y)$ 为该栅格点的静态水深值；$T(x,y,t)$ 为该栅格时刻 t 的潮位值；$\Delta(x,y,t)$ 为该格网点在时刻 t 影响水位变化的其他因素所产生的水深变化。$T(x,y,t)$ 与 $\Delta(x,y,t)$ 的和构成瞬时水深中的动态水深项。

1.4 安全水深区获取

考虑潮汐的瞬时水深会随时间变化，USV 还会受波浪、风等影响发生摇摆、升沉，带来吃水变化，从而使安全水深区和碍航区边界发生变化。根据 USV 安全航行所需的最小安全水深值，可以不断更新安全水深区。水深边界条件可以表示为：

$$f=S-H_{\min}+H_{\text{vertical}} \tag{7}$$

式中：$H_{\min}$ 为无人艇航行的最小安全水深值；H_{vertical} 为 USV 受波浪、风等影响发生摇摆、升沉带来的垂直于平均海面的距离，可由 USV 的姿态传感器获取(具体算法不在本文讨论范围)。

当 $f=0$ 时，即瞬时水深与最小安全水深相等。当 $f<0$ 时，瞬时水深小于最小安全水深值，即可以将此栅格判断为浅水碍航区，认为在此区域航行会发生危险。进行节点扩展时 USV 将选择水深值较大的栅格。通过动态水深值动态更新安全区域碍航区，从而进行全局的路径规划。

2 路径规划算法

2.1 D* Lite 算法

D* Lite 算法是在一种适用于动态环境的路径规划算法。在变化环境中出现新的未知障碍物时，采用该算法可以通过更新未知障碍物周围节点的信息，及时进行重新规划。

D* Lite 算法采用从目标节点到起始节点的扩展搜索方式，其代价函数 rhs(s)定义如下：

$$\text{rhs}(s)=\begin{cases}0, & s=s_{\text{goal}}\\ \min\limits_{s'\in \text{Succ}(s)} c(s,s')+g(s'), & \text{otherwise}\end{cases} \tag{8}$$

式中：Succ(s)表示当前节点 s 的子节点集；$c(s,s')$ 表示当前节点 s 到其子节点 s' 的距离代价值，本文采用欧式距离 $c(s,s')=\sqrt{(x_s-x_{s'})^2+(y_s-y_{s'})^2}$ 计算其代价；$g(s')$ 表示子节点 s' 到目标节点 s_{goal} 的移动距离代价。D* Lite 算法与静态规划算法的区别在于，D* Lite 算法拓展栅格同时记录 $g(s)$ 和 rhs(s) 两个变量。当 rhs(s) $\neq g(s)$ 时，表示通过该节点时处于不同状态，路径发生改变，则重新更新。当节点的两个变量相等时，即 rhs(s) $=g(s)$，表示节点拓展完成可以通行。文中定义启发函数为 $h(s)$，$h(s,s_{\text{start}})$ 为当前节点 s 与起始节点 s_{start} 的估计启发代价值。k_m 为 $h(s_{\text{last}},s_{\text{current}})$，$s_{\text{last}}$ 为检

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

测到的海图变化的节点，s_{current} 为当前检测到变化的节点，k_m 即为每一检测阶段移动距离之和。

$$h(s,s_{\text{start}})=\begin{cases}0, & s=s_{\text{start}}\\ c(s,u)+h(u,s_{\text{goal}}), & \text{otherwise}\end{cases} \tag{9}$$

在 D* Lite 算法中，存在不连续节点的优先队列，定义为 U，进入 U 中的节点按照二维向量 key(s)值排序，key(s) = $[k_1(s),k_2(s)]$，$k_1(s)$ 和 $k_2(s)$ 是优先队列排列参数。

$$k_1(s)=\min(g(s),\text{rhs}(s))+h(s_{\text{start}},s)+k_m \tag{10}$$

$$k_2(s)=\min(g(s),\text{rhs}(s)) \tag{11}$$

在路径规划过程中，D* Lite 算法采用八邻域的节点拓展方式，得到的路径并不平滑，与 USV 理想的航行路径差距较大。在动态海洋环境下，极有可能在水深较浅位置出现新障碍物带来航行危险。针对上述问题，本文提出 LT-D* Lite(light and depth-D* Lite)算法，进行安全平滑路径的规划。

2.2　LT-D* Lite 算法

2.2.1　视线检查算法

视线检查算法(line-of-sight，LOS)用于判断两点之间连线所经过的栅格中是否有障碍物存在。设两个栅格点的坐标分别为 (x_0,y_0) 和 (x_1,y_1)，dx 与 dy 为两栅格点横纵坐标之差，sx 与 sy 分别表示横纵坐标方向前进的步长和当前短轴与长轴的差。

LOS 检查算法中，分为 $dx<dy$ 和 $dy<dx$ 两种假设情况。本文以 $dx<dy$ 为例，部分流程图如图 1 所示，其中 Obs(x_s，y_s) 表示 s 点处为障碍物栅格。

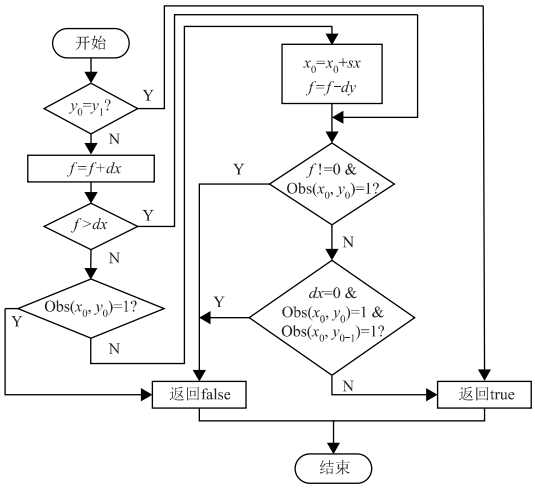


图 1　视线检查算法流程

Fig. 1　Flowchart of Line of Sight Checking Algorithm

在 D* Lite 算法中，当前节点 s 存在父节点，且均为相邻节点，定义为 parent(s)。LT-D* Lite 算法中，当前节点 s 的父节点为通过视线确定的视线父节点 lparent(s)，通过对节点进行视线检查而得到。lparent(s)的更新方式为：

$$\text{lparent}(s)=\begin{cases}\text{lparent}(s'), & \text{rhs}(s)>c(\text{lparent}(s'),s)+\text{rhs}(\text{lparent}(s'))\\ \text{parent}(s), & \text{NOT Line of Sight}(\text{lparent}(s),s)\\ \text{parent}(s), & \text{lparent}(s)=\text{NULL}\end{cases} \tag{12}$$

LT-D* Lite 算法中，确定视线父节点的示意图见图 2。$A4$ 节点为目标节点，$B3$ 节点为其邻域节点，故其视线父节点与邻域父节点相同。进一步拓展，如图 2 所示，$B2$ 节点与 $A4$ 节点之间存在障碍物，视线检查不成立，其不可为 $A4$ 节点的父节点，故其视线父节点仍为邻域节点。对于 $B3$ 节点与 $C1$ 节点之间视线检查成立，则其视线父节点不再是邻域节点。

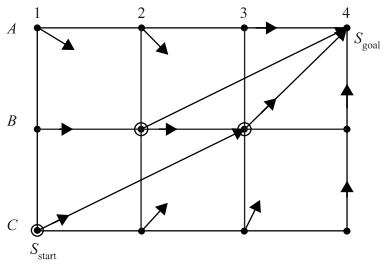


图 2　视线父节点示意图

Fig. 2　Schematic Diagram of the Line of Sight Parent Node

在 LT-D* Lite 算法中进行视线检查时，采用懒惰更新的方式，即假设节点进行视线检查成功，其父节点为视线父节点。当在优先队列中实际检查是否可视，若不可视，则重新设置父节点为相邻父节点。

2.2.2　路径水深危险度

全局路径规划在考虑 USV 航行路径长度时，要提高整体路径的安全性，本文在环境模型中引入水深危险度的指标，安全水深区中 USV 的最小安全水深值与环境水深值比例代表危险程度，定义为 $r(n)$，如式(13)所示。

$$r(n)=\frac{S_{\min}}{D(n)},(S_{\min}<D(n)) \tag{13}$$

水深危险度表示水下障碍物与 USV 的危险程度关系，可以为 USV 等移动机器人的路径规划提供准确数值依据，从而选择安全性更高的路径。

通过式(14)获得各栅格点的水深危险度后，$t(s_{i-1},s_i)$ 取从 s_{i-1} 到 s_i 的路径所途经栅格节点之间的危险度之和。

$$t(s_{i-1},s_i)=\sum_{j}^{n}r(s_i^j) \tag{14}$$

式中：$r(s_i^j)$ 为 i 路段中途经的第 j 个栅格节点的水

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

深危险度。节点 s_i^j 的确认方式借鉴 Bresenham 算法[13]。

$R_T(s)$ 是规划算法的路径中已经过路径的危害成本之和。

$$R_T(s) = \sum_i^n t(s_{i-1}, s_i) \tag{15}$$

在 D* Lite 规划算法中，根据距离代价值进行寻优，仅考虑避开障碍物，从而规划最佳路径。针对 USV 等受动态水深环境因素影响的路径规划，采用距离代价、水深危险度代价的综合评估，规划安全较优路径。与仅对障碍物进行规避的算法相比较，采用路径代价综合评估技术可以更好地适应规划区域的情况以及提高路径安全性的需求。

2.3 LT-D* Lite 算法流程

LT-D* Lite 算法的主函数伪代码如算法 1 所示。其中所用函数的伪代码见算法 2～算法 5，算法 2 为逐步规划函数，算法 3 为初始化函数，算法 4 为代价更新函数，算法 5 为优先级函数。

算法 1 Procedure Main.

```
function main
s_last = s_start
Initialize()
ComputeShortestPath()
while s_start ≠ s_goal do
s_start ← lparent(s_start)
Move to s_start
if any edge costs changed
 k_m = k_m + h(s_last, s_start)
 s_last = s_start
 for all directed edges u, v with changed
 edge costs do
Update edge cost c(u, v)
DistanceTrsnsformsLocal (v)
UpdateVertex(u)
end for
ComputeShortestPath()
end While
end function
```

算法 2 Procedure function.

```
function ComputeShortestPath
while U. TopKey() > CalculateKey
(s_start) or rhs(s_start) ≠ g(s_start) do
k_old ← U. TopKey()
u ← s which satisfy U. TopKey()
if NOT Line of Sight (lparent(u), u)
lparent(u) ← parent(u)
end if
if k_old < key(u) then
U. insert(s_goal)
 U. insert [Key(s_goal)]
 else if g(u) > rhs(u) then
 g(u) ← rhs(u)
 for s ∈ Pr ed(u) do
 UpdateVertex(s)
 if rp(u) and s ∈ U then
 if c(rp(u), s) + rhs(lparent(u)) < rhs(s)
 lparent(s) ← lparent(u)
 end if
 end if
 end for
 else g(s) ← ∞
 for s ∈ Pr ed(u) ⋃ u do UpdateVertex(s)
 end for
 end if
 end while
 end function
```

算法 3 Procedure function.

```
function Initialize
U ← ϕ
k_m ← 0
for s ∈ S do
rhs(s) ← ∞
g(s) ← ∞
end for
DistanceTransform(S)
rhs(s_goal) ← ∞
 parent(s_goal) ← s_goal
 lparent(s_goal) ← s_goal
 U. insert(s_goal)
 U. insert [Key(s_goal)]
 end function
```

算法 4 Procedure function.

```
function UpdateVertex(u)
if u ≠ s_goal then
rhs(s) ← min_{s' ∈ Succ(s)} (c(u, s') + g(s') + R_T(s') + t(u, s'))
parent(s) ← s' which satisfy min_{s' ∈ Succ(s)} (c(u, s') + g(s') + R_T(s') + t(u, s'))
end if
if lparent(u) == NULL then
lparent(u) ← parent(u)
end if
```

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

```
9   if u∈U then
10   U. Remove(u)
11   end if
12   if g(u)≠rhs(u) then
13   U. insert(u)
14   U. insert [Key(u)]
15   end if
16   end function
```

算法 5　Procedure function.

```
1   function Key(s)
2   return [min(g(s), rhs(s))+h(s_start, s)+
    k_m; min(g(s), rhs(s))]
3   end function
```

LT-D* Lite 算法首先进行预规划，预规划采用由目标节点向起始节点的反向搜索方式。此时沿着视线父节点进行回溯，即可寻得预规划路径。若在途中障碍物发生变化，需要进行重规划，在新障碍物处则将障碍物所对应的环境海图设置为障碍物空间，根据其为起点利用预规划路径信息重新规划出新的路径。

3　仿真实验

为了验证该模型及算法的有效性，下面进行仿真分析。为了确保比较结果的准确性，除了对比项有差异，其他类别都相同。所有实验测试均在 Intel Core I5-8400 CPU，2.80 GHz，8 GB RAM，Windows 10 上执行，路径规划算法在 Matlab2018 中实现。

3.1　仿真基础底图构建

为验证所提算法的有效性，本文以天津大学自主研发的“Dolphin I”无人艇为研究对象，其相关参数，如表 1 所示。

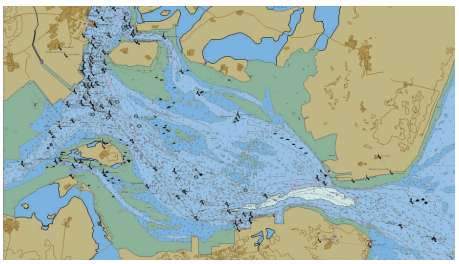
表 1　水深最小安全值的相关参数

Tab. 1　Related Parameters of the Minimum Safe Value of Water Depth

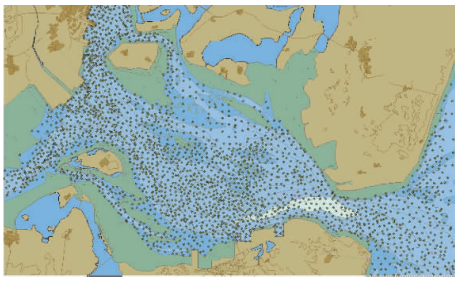
船长 L/m	船宽 D/m	最大航速 v/(m·s^{-1})	最大航时 t_{max}/h	最小安全水深值[4] H_{min}/m	回转半径[20] d/m
3.2	2.2	5	2	1.29	25.5

本文选择某海图作为基础底图进行分析，所包含区域涵盖陆地、岛屿和浅水区等基本的碍航物信息，如图 3 所示。

在对海图进行栅格化处理时需要确定栅格尺寸，栅格尺寸一般和无人艇大小为同一量级，受限于精度和尺度，需要综合起点到终点距离、无人艇大小和运动特性、定位精度等诸多因素。本文以“Dolphin I”无人艇为研究对象，其船长为 3.2 m，回转半径为 12.7 m，整个海图区域长宽为 26 041 m×18 190 m，定位精度可以合理控制在分米级，综合上述因素将实验栅格大小设置为 15 m×15 m。根据栅格大小进行水深插值处理，得到栅格化的水深数据(单位：m)，其静态环境模型如图 4 所示。



(a)基础底图



(b)离散水深点

图 3　S-57 电子海图环境

Fig. 3　S-57 Electronic Chart Marine Environment

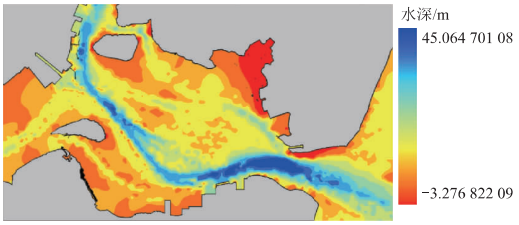


图 4　水深分布图

Fig. 4　Water Depth Interpolation

本文的潮汐数据采用该海域某验潮站发布的 2019 年潮汐表数据，潮汐表提供高低潮数据，插值得到逐时潮汐值。考虑到 USV 的航时，本文选取 24 h 的潮位数据，如图 5 所示。

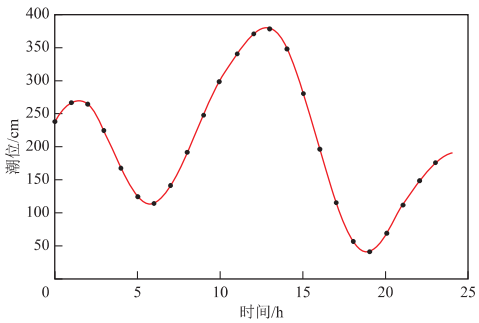


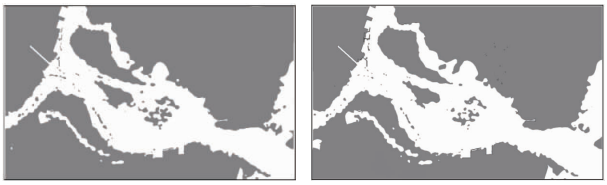
图 5　潮位值

Fig. 5　Tide Level Value

通过式(7)对图 4 所示基础底图的数据进行处理，即可建立瞬时水深模型，由于动态水位的影响，

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

瞬时水深和碍航区不断变化。当潮位值较高时(如在潮位最高时)，碍航区变化不明显，为了便于观察，本文选取潮位值为 0.42 m(19 h)和 1.13 m(21 h)的瞬时水深和碍航区进行比对，如图 6 所示。



(a)低潮位环境图 (b)高潮位环境图

图 6 高低潮位碍航区

Fig. 6 Obstruction Area under High and Low Tide

3.2 实验结果及分析

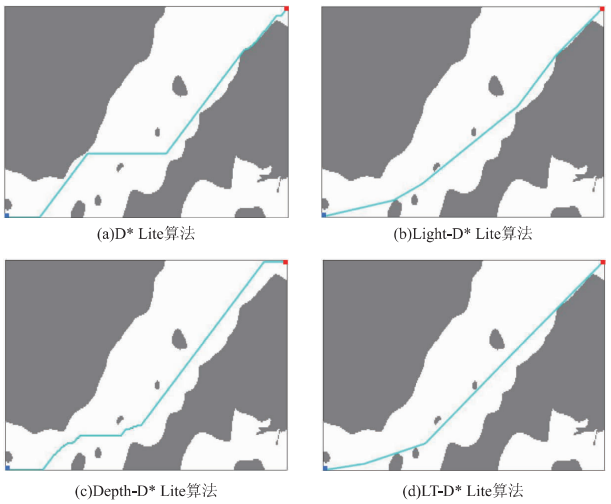
在实验中，与其他文献一样，遵循如下假设：①USV 可以安全地通过可航行区域的网格顶点；②在路径规划过程中，暂时忽略航行过程中可能遇到的运动障碍；③USV 在进行总体规划时简化为二维规划。

本文采用规划的路径长度、路径水深危险度和算法规划效率 3 种评价指标对路径质量进行评估。路径长度的比较是基于每个相邻路径节点的欧氏距离之和。规划算法的运行效率为程序运行效率，对于启发式规划算法，其拓展节点数与运行效率成正相关。规划路径所经过的栅格危险度之和为整体路径危险度，可由式(15)表示。

3.2.1 预规划路径

本文选取潮位值为 1.13 m(21 h)的瞬时水深模型为预规划环境模型。在所构建的栅格海图上，起点设置为(110.363°E，21.098°N)，用蓝色点表示，终点为(110.430°E，21.117°N)，用红色点表示，分别使用 D* Lite 算法、引入视线检查算法的 D* Lite 算法(Light-D* Lite 算法)、考虑水深危险的 D* Lite 算法(Depth-D* Lite 算法)以及本文构建的 LT-D* Lite 算法进行仿真，得到如图 7 所示的预规划路径。

图 7(a)为 D* Lite 算法规划所得的路径，图 7(b)为只利用视线检查算法和懒惰更新法改进 D* Lite 算法，规划出的路径相对于图 7(a)的路径较平滑。图 7(c)为 D* Lite 算法与水深危险的结合，规划出的路径为寻找水深值较高、较安全的路径，但是整体路径较不平滑。图 7(d)为本文 LT-D* Lite 算法规划出的路径，路径整体较平滑。由表 2 中的数据可以看出，LT-D* Lite 算法综合其余两种算法的优点，与 D* Lite 算法相比，整体路径长度减少 4%，同时路径安全性也提高 14.36%。路径拓展节点数相对于 D* Lite 算法有增加，表明引入视线检查算法和水深危险度，消耗的时间有所增加。



(a)D* Lite算法 (b)Light-D* Lite算法

(c)Depth-D* Lite算法 (d)LT-D* Lite算法

图 7 预规划路径

Fig. 7 Results of Pre-planned Path

表 2 路径规划指标

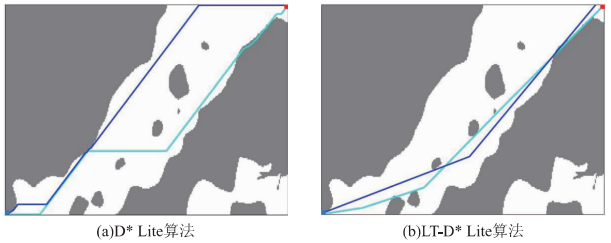
Tab. 2 Indicators of Path Planning

规划算法	路径长度/m	拓展节点数/个	路径危险度
D* Lite 算法	5 428	12 088	149.7
Light-D*-Lite 算法	5 180	16 565	221.6
Depth-D* Lite 算法	5 501	18 106	106.2
LT-D* Lite 算法	5 191	18 338	128.2

3.2.2 路径重规划

在进行预规划时，由于时间推移潮位变化导致碍航区分布发生变化，原栅格环境海图亦发生相应变化。如当落潮时，安全水深区变成碍航区，沿原路径航行的过程中产生新的障碍物。为了清晰观察重规划路径的变化，以潮位值为 0.42 m 的瞬时水深环境为重规划环境，在障碍物变化后的环境模型中显示预规划路径和重规划路径，浅蓝色代表预规划路径，深蓝色代表重规划路径。

图 8(a)为传统 D* Lite 的重规划结果，图 8(b)为本文算法进行路径重规划的结果，重规划之后的路径成功避开新的障碍物。当考虑水深危险因素，算法会寻找水深值较大区域，而不是仅沿着障碍物进行规划。



(a)D* Lite算法 (b)LT-D* Lite算法

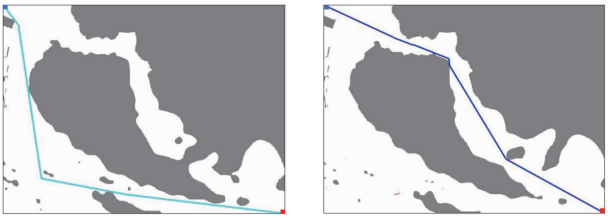
图 8 重规划路径

Fig. 8 Results of Replanned Path

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

3.2.3　狭窄水道路径规划

在上述例子中，选择以落潮情况进行案例分析，得出 LT-D* Lite 算法可以合理避开障碍物。为了体现算法可以合理规划路径，选择以低潮位图为预规划图，以高潮位图为重规划图，起点为(110.406°E，21.179°N)，终点为(110.495°E，21.122°N)，其规划路径如图 9 所示。



(a)低潮位　　(b)高潮位

图 9　狭窄水道路径

Fig. 9　The Path in Narrow Waterway

由图 9 可以看出，当在潮位值较低时，障碍物较多，在狭窄水道区域出现碍航区，规划路径无法通过，当潮位值升高后，碍航区减少，原狭窄水道的水深值增大，确认为安全水深区，此时 USV 重规划路径可以通过。

3.2.4　算法对比

为了验证算法的可靠性和一致性，本文选取6 种不同的场景进行对比分析。通过算法对比可以观察本文 LT-D* Lite 算法的优异性，本文选取随机采样算法-RRT* 算法作为对比算法[21]。本节分别用 D* Lite 算法、RRT* 算法以及改进的 LT-D* Lite 算法进行路径规划仿真，并且从路径长度、拓展节点数、路径危险度 3 个指标对规划结果进行对比。规划结果如图 10，其中橙色代表 D* Lite 算法，绿色代表 RRT* 算法，紫色代表本文的 LT-D* Lite 算法。

如图 10(a)所示，在规划时间方面，本文提出的 LT-D* Lite 算法普遍高于 D* Lite 算法，这是因为 LT-D* Lite 算法考虑水深危险度，拓展节点在一定程度上有所增加，同时需要对拓展节点进行视线检查运算，运算量有所增加。RRT* 算法的规划时间远大于其余两种算法，此算法为了保证路径的渐进最优性，需要消耗较长时间。在二维环境中，此类基于采样的随机算法具有较大劣势，其更适用于高维空间中的规划。尤其在动态环境下，D* Lite 算法和本文的 LT-D* Lite 算法具备增量规划，可以实现较少的重规划次数，从而较少影响节点数，优化重规划时间。RRT* 算法等普通适用于静态环境的路径规划算法，对于动态环境进行重新规划，均需对海图进行重新搜索，消耗时间较长。

由图 10(b)可以看出，在路径长度上，本文的

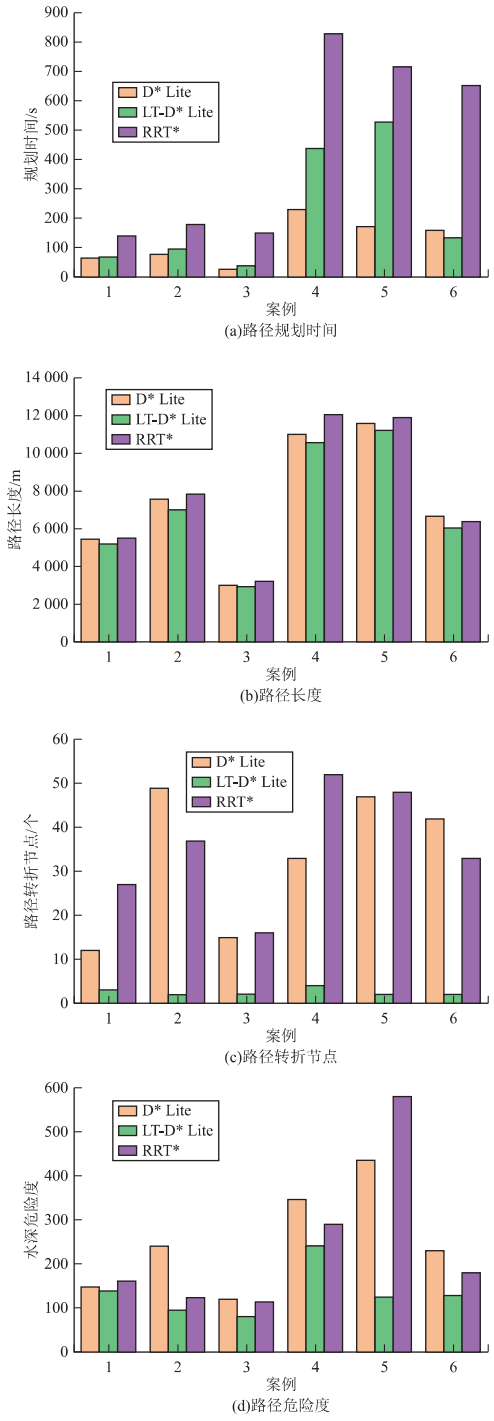


图 10　3 种算法规划路径参数对比

Fig. 10　Comparison of Three Algorithms

LT-D* Lite 算法处于优势。LT-D* Lite 算法利用视线检查算法进行关键节点的直接相连，有效缩短路径距离，对路径长度有较大的缩减，反映了算法改进的优越性。

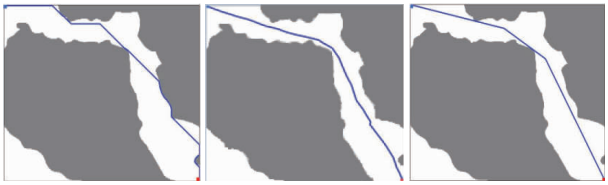
在路径转折节点，如图 10(c)所示，本文的 LT-D* Lite 算法表现明显优于其他两类算法，约有 3～5 个节点。原始的 D* Lite 算法与 RRT* 算法的节点数均大于 10 个，对于 USV 等路径规划后的行动十分不利。上述现象的原因是 D* Lite 算

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

法将相邻节点作为父节点，在障碍物等拐角处节点会产生截断，导致路径转折节点增加，产生较多弯折，不符合 USV 等航行器的运动。同时 RRT* 算法是基于随机采样，虽然相对于普通 RRT 算法，RRT* 算法进行改进，其路径为渐进最优，整体路径平滑性有较大程度上的提高，但是其路径节点仍然具有随机性，尤其是通过狭窄通道时，会产生较多的随机点，使路径转折节点增多，与本文利用视线检查算法改进后还是有较大程度上的差距。LT-D* Lite 算法对于障碍物周围选取较少的关键节点，使得路径关键节点直连，路径更加平滑，更符合 USV 等的运动。

在路径危险度上，D* Lite 算法和 RRT* 算法均未考虑水深影响因素，其在规划过程中仅以启发值作为代价函数，即仅考虑路径距离，导致路径极易穿越水深较浅危险区域，整体路径危险度较高。本文的 LT-D* Lite 算法考虑水深危险代价后，穿过水深值较大区域，整体路径危险度相对小，尤其在进行大区域规划时，更能体现路径危险性差距，保证 USV 等在港口水域的航行安全性。

图 11 是从上述比较的 6 种案例中随机挑选一种案例进行对比，其起点为(110.471°E，21.137°N)，终点为(119.471°E，21.137°N)，3 种算法均规划出一条路径，从图 11 中可以看出 D* Lite 算法和 RRT* 算法的路径均存在转折较多的特点，LT-D* Lite 算法规划的路径转折点仅有 2 个。RRT* 算法在规划过程中，采用随机采样点进行路径节点的拓展，路径节点的选择没有根据精确的数值比较，因此规划的路径存在一定的随机性，导致规划的路径较冗长，且最终路径的生成是对较优的随机采样点进行连接，因此存在转折点较多的缺点，不如本文的 LT-D* Lite 算法。本文 LT-D* Lite 算法的路径，有平滑、安全，且路径较短等优点，但是同时也存在一定的缺陷，其规划路径拓展节点较多，所用时间较长。



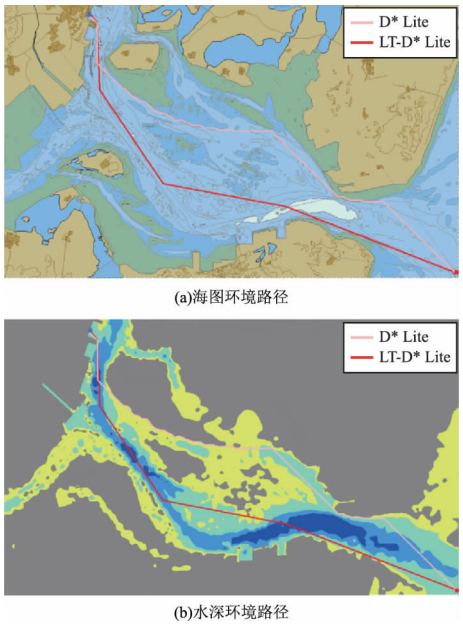
(a)D* Lite算法 (b)RRT*算法 (c)LT-D* Lite算法

图 11 3 种算法规划结果

Fig. 11 Planning Results of Three Algorithms

3.3 复杂环境下的规划

为显示 USV 在港口水域的路径，在所选海图环境中进行路径规划，起点为(110.406°E，21.179°N)，终点为(110.594°E，21.053°N)，如图 12 所示，粉色代表 D* Lite 算法规划的路径，红色代表 LT-D* Lite 算法规划的路径。如图 12(a)所示，传统 D* Lite 算法规划出的路径与海滩地带十分接近；图 12(b)可以看出当潮位值较低时，这些区域为浅水深区域，路径的水深危险度较高。从图 12(b)可以看出，本文 LT-D* Lite 算法规划出的路径沿着水深值较大的区域航行，路径安全性较高，且平滑程度也较高。



(a)海图环境路径

(b)水深环境路径

图 12 港口区域长距离路径规划

Fig. 12 Long Distance Path Planning in Port Area

4 结束语

本文采用 D* Lite 算法对 USV 进行动态规划，并针对其不适用动态海洋变化环境和路径不平滑的问题进行改进，提出了一种 LT-D* Lite 算法。该算法通过建立水深危险代价使得其规划路径选择水深较安全区域，同时引入视线检查算法和懒惰更新算法平滑路径。本文将改进后的 LT-D* Lite 算法进行仿真分析，验证算法在动态环境中的适应性，并与传统 D* Lite 算法和随机采样 RRT* 算法进行对比，对比表明本文所提出的算法的路径具有长度较短、平滑度高、安全性高的优点。然而，由于视线检查算法带来的大量节点的检查，导致规划算法时间消耗较大，在今后将进一步提高算法的规划效率。

参考文献

[1] LIU Y C, BUCKNALL R, ZHANG X Y. The fast marching method based intelligent navigation of an unmanned surface vehicle [J]. Ocean Engineering,

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

2017,142:363-376.

[2] 顾尚定,周春辉,文元桥,等.基于拓扑位置关系的无人艇路径搜索方法[J].中国航海,2019,42(2):52-58.(GU Shangding, ZHOU Chunhui, WEN Yuanqiao, et al. Path search of unmanned surface vehicle based on topological location[J]. Navigation of China, 2019, 42(2):52-58.)

[3] 孔德鑫,刘洋,赵金,等.密集岛礁环境下的无人艇的路径规划[J].信息通信,2020,33(3):14-17.(KONG Dexin, LIU Yang, ZHAO Jin, et al. Unmanned surface vehicle path planning in dense island reef environment[J]. Information & Communications, 2020, 33(3):14-17.)

[4] LIU Shuai, WANG Chenxu, ZHANG Anmin. A method of path planning on safe depth for unmanned surface vehicles based on hydrodynamic analysis[J]. Applied Sciences, 2019, 9(16):3228.

[5] ZHOU Chunhui, GU Shangding, WEN Yuanqiao, et al. The review unmanned surface vehicle path planning: based on multi-modality constraint[J]. Ocean Engineering, 2020, 200:107043.

[6] 林巍凌.引入导航网格的室内路径规划算法[J].测绘科学,2016,41(2):39-43.(LIN Weiling. Indoor path planning algorithm based on navigation mesh[J]. Science of Surveying and Mapping, 2016, 41(2):39-43.)

[7] NASH A, KOENIG S, TOVEY C. Lazy Theta*: any-angle path planning and path length analysis in 3D[C]// Proceedings of the Twenty-fourth AAAI Conference on Artificial Intelligence. [S. l.]: [s. n.], 2010:147-154.

[8] STENTZ A. Optimal and efficient path planning for partially-known environments[C]// Proceedings of the 1994 IEEE International Conference on Robotics and Automation. [S. l.]: IEEE, 1994:3310.

[9] LIKHACHEV M, KOENING S. A generalized framework for lifelong planning A* search[J]. Robotics and Autonomous, 2011, 5(59):329-342.

[10] KOENIG S, LIKHACHEV M. Fast replanning for navigation in unknown terrain[J]. IEEE Transactions on Robotics, 2005, 21(3):354-363.

[11] 张浩,孙新柱.增强 D* Lite 在自主移动机器人安全路径规划中应用[J].河北工程大学学报(自然科学版),2014,31(2):89-92.(ZHANG Hao, SUN Xinzhu. Enhanced D* Lite for safe path planning of autonomous mobile robot[J]. Journal of Hebei University of Engineering(Natural Science Edition), 2014, 31(2):89-92.)

[12] 张晓冉,居鹤华.采用改进 D* Lite 算法的自主移动机器人路径规划[J].计算机测量与控制,2011,19(1):155-157.(ZHANG Xiaoran, JU Hehua. Developed D* Lite algorithm for path planning of autonomous mobile robots[J]. Computer Measurement & Control, 2011, 19(1):155-157.)

[13] BRESENHAM J E. Algorithm for computer control of a digital plotter[J]. IBM Systems Journal, 1965, 4(1):25-30.

[14] FERGUSON D, STENTZ A. Field D*: an interpolation-based path planner and replanner[C/OL]. [2021-04-13]. http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=3A32450EC7E3EECB1BF2E9D2969E22FF?doi=10.1.1.72.8158&rep=rep1&type=pdf.

[15] 陈轶,彭认灿,郭立新,等.顾及潮汐变化的动态可航水域表示模型及方法研究[J].测绘科学,2009,34(1):38-40.(CHEN Yi, PENG Rencan, GUO Lixin, et al. Research on real-time visualization model and technique of navigable area with regard to tide[J]. Science of Surveying and Mapping, 2009, 34(1):38-40.)

[16] 王海波,张汉武,郝勇帅.横轴墨卡托海图的船舶极区航行方法[J].测绘科学,2018,43(5):149-154.(WANG Haibo, ZHANG Hanwu, HAO Yongshuai. Exploration to a sailing for ships in polar regions based on transverse Mercator chart[J]. Science of Surveying and Mapping, 2018, 43(5):149-154.)

[17] 徐琳.基于 ArcGIS 的地面沉降数据空间分析技术与应用[D].北京:中国地质大学(北京),2013.(XU Lin. Spatial analysis techniques and application of land subsidence data based on ArcGIS[D]. Beijing: China University of Geosciences, 2013.)

[18] 程明华.基于 GIS 的地层产状空间插值方法研究[D].北京:中国地质大学(北京),2010.(CHENG Minghua. Method research of spatial interpolation for the stratum attitude based on GIS[D]. Beijing: China University of Geosciences, 2010.)

[19] 杜佳芸.高精度动态水深模型与服务研究[D].天津:天津大学,2017.(DU Jiayun. Research on high resolution dynamic depth model and service[D]. Tianjin: Tianjin University, 2017.)

[20] ZHOU Jian, WANG Chenxu, ZHANG Anmin. A COLREGs-based dynamic navigation safety domain for unmanned surface vehicles: a case study of Dolphin-I[J]. Journal of Marine Science and Engineering, 2020, 8(4):264.

[21] KARAMAN S, FRAZZOLI E. Sampling-based algorithms for optimal motion planning[J]. International Journal of Robotics Research, 2011, 30(7):846-894.

(责任编辑:侯琳)

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net