

赞同 87

分享

EGO-Planner论文阅读笔记

CK1201
运动规划

87 人赞同了该文章

如果侵权，请联系我立刻删除。ckwu1201@163.com

欢迎交流。

论文：

EGO-Planner: An ESDF-free Gradient-based Local Planner for Quadrotors
arxiv.org/abs/2008.08835

代码：

EGO-Planner
github.com/ZJU-FAST-Lab/ego-planner

代码阅读笔记：

EGO-Swarm在EGO-Planner的基础上做了改进，例如可以选择从障碍物两侧飞过，选择最优的路径执行、去中心化等。

CK1201：EGO-Swarm代码阅读笔记汇总-FAST-Lab RA作业
53 赞同 · 11 评论 文章



摘要

1. 通过比较碰撞轨迹与无碰撞引导路径，得到惩罚函数中的碰撞项
2. 轨迹优化器只提取当前轨迹撞到的障碍物的信息
3. 如果某段轨迹动力学不可行，则延长该段轨迹分配的时间
4. 异性曲线拟合算法——在保持原有轨迹形状的情况下降低轨迹的阶数

一、介绍

ESDF的构建方式有全局增量式和批量本地计算两种方式，但他们并不是专门用于运动规划而构建的，也就是说对于运动规划来说现有的两种方法构建出的ESDF地图是多余的、不必要的。从图1中可以看出轨迹仅仅覆盖小范围的ESDF地图，大部分都是没用的。简单地手动减小ESDF地图范围，缺乏理论依据，也包含不必要的计算。

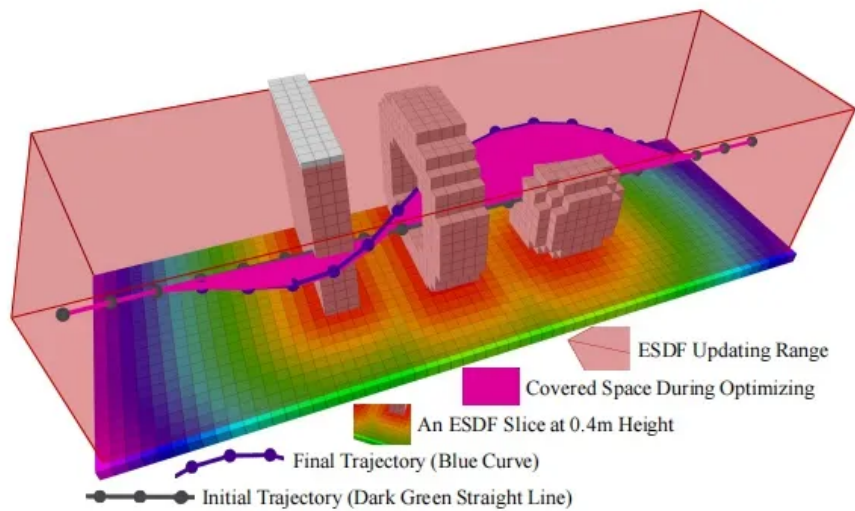


Fig. 1: Trajectory during optimizing just covers a very limited space of the ESDF updating range.

知乎 @CK1201

EGO-Planner主要由基于梯度的样条曲线优化器和细化过程组成。

1.1 基于梯度的样条曲线优化器

使用平滑性、碰撞性和动力学可行性项优化轨迹。碰撞项的构成通过比较障碍物内的轨迹与无碰撞的引导路径，然后用梯度信息将碰撞到障碍物的轨迹拉出障碍物，从而算法只需要计算碰撞处的障碍物梯度即可。

1.2 细化过程

当某段轨迹动力学不可行时，激活细化过程，即增大该轨迹分配的时间。新生成的B样条曲线平衡了动力学可行性与拟合之前动力学不可行轨迹的准确性。在轴向和径向上拟合的准确性惩罚并不一样，以提高模型的鲁棒性。

二、相关工作

2.1 基于梯度的运动规划

2.2 带符号的欧式距离场 (ESDF)

三、碰撞力的估计

定义B样条曲线的控制点为 Q

一开始生成一条满足终端约束但不考虑障碍物的B样条曲线 Φ 。

接着，对于每段被检测到的碰撞的线段，用优化程序（这里使用的是A*算法）生成一个无碰撞的路径 Γ 。对于发生碰撞的线段的每个控制点 Q_i 都会生成一个在障碍物表面的定位点（anchor

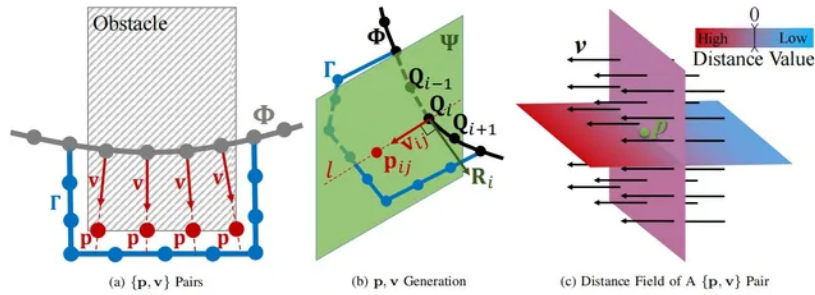


Fig. 3: a) A trajectory Φ passing through an obstacle generates several $\{p, v\}$ pairs for control points. p are the points at the obstacle surface and v are unit vectors pointing from control points to p . b) A plane Ψ which is perpendicular to a tangent vector R_i intersects Γ forming a line l , from which a $\{p, v\}$ pair is determined. c) Slice visualization of distance field definition; the color indicates the distance and the arrows are identical gradients equal to v . p is at the zero distance plane.

Algorithm 1 CheckAndAddObstacleInfo

```

1: Notation: Environment  $\mathcal{E}$ , Control Points Struct  $Q$ , Anchor Points  $p$ , Repulsive Direction Vector  $v$ , Colliding Segments  $S$ 
2: Input:  $\mathcal{E}$ ,  $Q$ 
3: for  $Q_i$  in  $Q$  do
4:   if FindConsecutiveCollidingSegment( $Q_i$ ) then
5:      $S$ .push_back(GetCollisionSegment())
6:   end if
7: end for
8: for  $S_i$  in  $S$  do
9:    $\Gamma \leftarrow \text{PathSearch}(\mathcal{E}, S_i)$ 
10:  for  $S_i.\text{begin} \leq j \leq S_i.\text{end}$  do
11:     $\{p, v\} \leftarrow \text{Find\_p\_v\_Pairs}(Q_j, \Gamma)$ 
12:     $Q_j.\text{push\_back}(\{p, v\})$ 
13:  end for
14: end for

```

知乎 @CK1201

Q_i 到第 j 个障碍物的距离为

$$d_{ij} = (Q_i - p_{ij})v_{ij} \quad (1)$$

为了防止轨迹被拉出当前障碍物前、迭代过程中反复生成 $\{p, v\}$ 对，判断是否为新障碍物的标准是：如果控制点 Q_i 处于障碍物中时，并且对于当前得到的所有障碍物 j 满足 $d_{ij} > 0$ ，则该障碍物为新发现的障碍物。从而只计算影响轨迹的障碍物信息，减少运行时间。

如图2，由于不足和错误的ESDF信息会导致规划器落入局部最优解。一个无碰撞初始轨迹 (A^*) 能够在避免碰撞上提供比ESDF更好的信息，因为明确设计的排斥力对各种任务和环境可以相当有效。

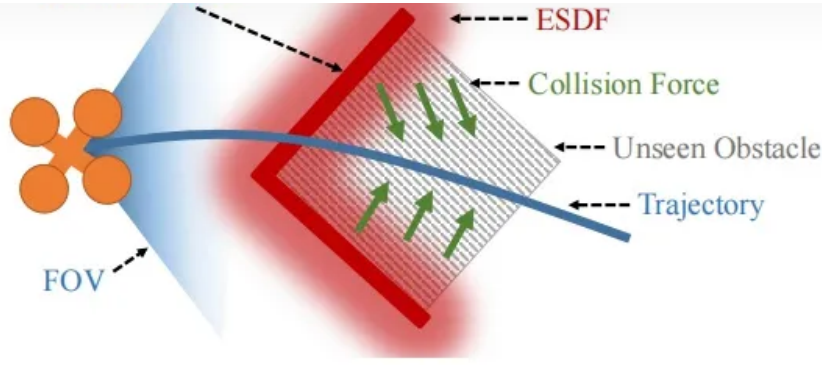


Fig. 2: The trajectory gets stuck into a local minimum, which is very common since the camera has no vision of the back of the obstacle.

知乎 @CK1201

四、基于梯度的轨迹优化器

4.1 问题建模

本文使用均匀B样条曲线 Φ 来表示轨迹，它由阶数 p_b 、 N_c 个控制点 $\{Q_1, Q_2, \dots, Q_{N_c}\}$ 和一个节点向量 $\{t_1, t_2, \dots, t_M\}$ 决定，其中 $Q_i \in \mathbb{R}^3$ ， $t_m \in \mathbb{R}$ ， $M = N_c + p_b$ 。（均匀B样条曲线可以参考计算机图形学bezier曲线曲面B样条曲线曲面）均匀B样条曲线的每个节点有相同的时间间隔 $\Delta t = t_{m+1} - t_m$ 。

B样条曲线的凸包性质表明，某段曲线只与 $p_b + 1$ 个连续的控制点有关，并且曲线被包含在这些点构成的凸包内。B样条曲线的k阶导数是 $p_b - k$ 阶B样条曲线。轨迹 Φ 的一阶、二阶、三阶导数的控制点分别为

$$V_i = \frac{Q_{i+1} - Q_i}{\Delta t}, A_i = \frac{V_{i+1} - V_i}{\Delta t}, J_i = \frac{A_{i+1} - A_i}{\Delta t} \quad (2)$$

根据无人机的微分平坦特性降低要规划的变量。优化问题可以被定义为

$$Q = \operatorname{argmin} J = \lambda_s J_s + \lambda_c J_c + \lambda_d J_d \quad (3)$$

J_s 为光滑项， J_c 为碰撞项， J_d 为动力学可行项， $\lambda_s, \lambda_c, \lambda_d$ 为惩罚项的权值。

4.1.1 光滑项惩罚

最小化高阶导数能够使得整段轨迹光滑。所以光滑项由加速度和加加速度的平方和构成。由于B样条曲线的凸包性质，只要最小化轨迹 Φ 的二阶和三阶控制点的平方和就能够有效地减小加速度和加加速度的平方和。

$$J_s = \sum_{i=1}^{N_c-2} \|A_i\|_2^2 + \sum_{i=1}^{N_c-3} \|J_i\|_2^2 \quad (4)$$

4.1.2 碰撞项惩罚

通过惩罚 $d_{ij} < s_f$ 安全界限的控制点，将控制点推离障碍物。为了便于优化，我们构造了一个二阶连续的惩罚函数 j_c ，随着 d_{ij} 的减小， j_c 的斜率就越小。

$$j_c(i, j) = \begin{cases} 0, & c_{ij} \leq 0 \\ c_{ij}^3, & 0 \leq c_{ij} \leq s_f \\ 3s_f c_{ij}^2 - 3s_f^2 c_{ij}, & \end{cases} \quad (5)$$

Q_i 的 $\{p, v\}_j$ 对个数。对所有控制点的cost求和得到总的碰撞项

$$J_c = \sum_{i=1}^{N_c} j_c(Q_i) \quad (6)$$

相比于传统用三线性插值的方法求碰撞项的梯度，我们直接闭式计算 J_c 对 Q_i 的导数

$$\frac{\partial j_c}{\partial Q_i} = \sum_{i=1}^{N_c} \sum_{j=1}^{N_p} \begin{cases} 0, & c_{ij} \leq 0 \\ -3c_{ij}^2, & 0 \leq c_{ij} \leq s_f \\ -6s_f c_{ij} + 3s_f^2, & c_{ij} > s_f \end{cases} \quad (7)$$

4.1.3可行项惩罚

动力学可行项通过限制轨迹在三个轴上的k阶导数来实现。得益于B样条曲线的凸包性质，只要对控制点进行限制即可满足动力学可行性。

$$J_d = \sum_{i=1}^{N_c-1} \omega_v F(V_i) + \sum_{i=1}^{N_c-2} \omega_a F(A_i) + \sum_{i=1}^{N_c-3} \omega_j F(J_i) \quad (8)$$

$$F(C) = \sum_{r=x,y,z} f(c_r) \quad (9)$$

$$f(c_r) = \begin{cases} a_1 c_r^2 + b_1 c_r + c_1, & c_r < -c_j \\ (-\lambda c_m - c_r)^3, & -c_j < c_r < -\lambda c_m \\ 0, & -\lambda c_m \leq c_r \leq \lambda c_m \\ (c_r - \lambda c_m)^3, & \lambda c_m < c_r < c_j \\ a_2 c_r^2 + b_2 c_r + c_2, & c_r > c_j \end{cases} \quad (10)$$

其中， $\omega_v, \omega_a, \omega_j$ 为每项的权重， $c_r \in C \in \{V_i, A_i, J_i\}$ ， $a_1, b_1, c_1, a_2, b_2, c_2$ 用来满足函数二阶连续性， c_m 导数限制， c_j 为二次和三次函数的交界， $\lambda < 1 - \epsilon$ ($\epsilon \ll 1$) 为弹性系数，使得最终的结果满足约束。

4.2数值优化

目标函数 J 会随着新障碍物的加入而不断改变，这就要求求解器能够快速重启，并且目标函数主要由二次项组成，所以Hessian信息能够加快收敛速度。但得到精确的Hessian消耗大量计算机资源。所以我们使用拟牛顿法（quasi-Newton methods）从梯度信息中来近似计算Hessian。

在对比了Barzilai-Borwein method、truncated Newton method和L-BFGS method后，L-BFGS表现最好，平衡了重启损失和逆Hessian估计的准确性。L-BFGS从以前的目标函数评估近似Hessian，但需要一系列的迭代，以达到一个相对准确的估计。

对于一个无约束的优化问题 $\min_{x \in \mathbb{R}^n} f(x)$ ， x 的更新近似牛顿法

$$x_{k+1} = x_k - \alpha_k H_k \nabla f_k \quad (11)$$

$$H_{k+1} = V_k^T H_k V_k + \rho_k s_k s_k^T \quad (12)$$

其中 α_k 为步长， $\rho_k = (y_k^T s_k)^{-1}$ ， $V_k = I - \rho_k y_k s_k^T$ ， $s_k = x_{k+1} - x_k$ ， $y_k = \nabla f_{k+1} - \nabla f_k$ 。

算法将式（12）右乘 ∇f_k ，递归扩展m步后得到了有效的、线性时间/空间复杂度的双环递归更新方法。L-BFGS的始化逆Hessian H_k^0 由Barzilai-Borwein step的权重来得到

$$H_k^0 = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}} I$$

五、时间重分配和轨迹细化

基于四中得到的安全轨迹 Φ_s ，生成一条时间分配合理的均匀B样条曲线轨迹 Φ_f ，然后使用各向异性曲线拟合方法（an anisotropic curve fitting method）使 Φ_f 自由地优化其控制点，以满足高阶导数约束，同时保持与 Φ_s 几乎相同的导数形状。

首先计算超过限制的比例

$$r_e = \max\{|V_{i,r}/v_m|, \sqrt{|A_{j,r}/a_m|}, \sqrt[3]{|J_{k,r}/j_m|}, 1\} \quad (14)$$

其中 $i \in \{1, \dots, N_c - 1\}$, $j \in \{1, \dots, N_c - 2\}$, $k \in \{1, \dots, N_c - 3\}$, $r \in \{x, y, z\}$ 。 r_e 表明相对于 Φ_s ， Φ_f 需要多分配多少时间。 V_i ， A_j 和 J_k 分别与 Δt 的一次、二次和三次成反比。则 Φ_f 的新时间间隔为

$$\Delta t' = r_e \Delta t \quad (15)$$

通过求解一个闭式的最小最小二乘问题，在约束下初始生成时间跨度为 $\Delta t'$ 的轨迹 Φ_f ，同时保持与 Φ_s 相同的形状和控制点数。然后重新计算光滑项和可行项得到新的目标函数

$$Q = \operatorname{argmin} J' = \lambda_s J_s + \lambda_d J_d + \lambda_f J_f \quad (16)$$

其中 λ_f 是拟合项的权重（ Φ_f 与 Φ_s 的拟合程度）。

J_s 被定义为从 $\Phi_f(\alpha T')$ 到 $\Phi_s(\alpha T)$ 各向异性位移的积分，其中 T 和 T' 为轨迹 Φ_s 和 Φ_f 的时长， $\alpha \in [0, 1]$ 。由于拟合的曲线 Φ_s 已经无碰撞，对于两条曲线，我们用带有权重的轴向位移来放宽光滑调整限制，用高权重的径向位移来防止碰撞。如图5所示，我们使用球状度量来使在同一球体表面的位移产生相同的惩罚。

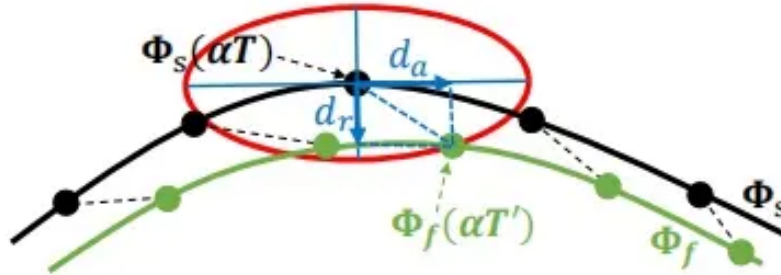


Fig. 5: Optimizing trajectory Φ_f to fit trajectory Φ_s while adjusting smoothness and feasibility. Black and green dots are sample points on the trajectory. The displacement between $\Phi_f(\alpha T')$ and $\Phi_s(\alpha T)$ breaks down into d_a and d_r along two ellipse principal axes. Points at the red ellipse surface produce identical penalties. 知乎 @CK1201

用于 $\Phi_f(\alpha T')$ 的椭圆体是一个以 $\Phi_s(\alpha T)$ 为中心的椭圆，绕它的主轴或切线 $\dot{\Phi}_s(\alpha T)$ 旋转一定角度得到的。则轴向位移 d_a 和径向位移 d_r 为

$$d_a(\alpha T') = (\Phi_f(\alpha T') - \Phi_s(\alpha T)) \cdot \frac{\dot{\Phi}_s(\alpha T)}{\|\dot{\Phi}_s(\alpha T)\|} \quad (17)$$

$$d_r(\alpha T') = \left\| (\Phi_f(\alpha T') - \Phi_s(\alpha T)) \times \frac{\dot{\Phi}_s(\alpha T)}{\|\dot{\Phi}_s(\alpha T)\|} \right\|$$

则拟合项为

$$J_f = \int_0^1 \left[\frac{d_a(\alpha T')^2}{a^2} + \frac{d_r(\alpha T')^2}{b^2} \right] d\alpha \quad (18)$$

其中 a 和 b 分别是椭圆的半长轴和半短轴。

6.1 实验细节

规划算法框架如算法2所示。

Algorithm 2 Rebound Planning

```
1: Notation: Goal  $\mathcal{G}$ , Environment  $\mathcal{E}$ , Control Point Struct  
    $\mathbf{Q}$ , Penalty  $J$ , Gradient  $\mathbf{G}$   
2: Initialize:  $\mathbf{Q} \leftarrow \text{FindInit}(\mathbf{Q}_{last}, \mathcal{G})$   
3: while  $\neg \text{IsCollisionFree}(\mathcal{E}, \mathbf{Q})$  do  
4:    $\text{CheckAndAddObstacleInfo}(\mathcal{E}, \mathbf{Q})$   
5:    $(J, \mathbf{G}) \leftarrow \text{EvaluatePenalty}(\mathbf{Q})$   
6:    $\mathbf{Q} \leftarrow \text{OneStepOptimize}(J, \mathbf{G})$   
7: end while  
8: if  $\neg \text{IsFeasible}(\mathbf{Q})$  then  
9:    $\mathbf{Q} \leftarrow \text{ReAllocateTime}(\mathbf{Q})$   
10:   $\mathbf{Q} \leftarrow \text{CurveFittingOptimize}(\mathbf{Q})$   
11: end if  
12: return  $\mathbf{Q}$ 
```

知乎 @CK1201

我们设置B样条曲线的阶数 $p_b=3$ ，控制点的个数 N_c 在25上下，具体由规划预期距离（大约7m）和初始的邻近点间距（大约0.3m）决定。这些参数根据经验，平衡了问题的复杂度和自由度得到。算法的时间复杂度为 $O(N_c)$ ，因为根据B样条曲线的性质，一个控制点只影响周围的几段轨迹。L-BFGS的复杂性在相同的相对公差上也是线性的。对于生成无碰撞的轨迹，我们使用A*算法，它生成的轨迹常常贴着障碍物。因此，我们可以直接在轨迹上选择定位点（anchor point） p 而不用搜索障碍物的表面。对于图3b中定义的向量 R_i ，通过均匀b样条参数化的性质，可以推导出

$$R_i = \frac{Q_{i+1} - Q_{i-1}}{2\triangle t} \quad \text{tag{19}}$$

式18被离散化为有限个数的点 $\Phi_f(k\Delta t)$ 和 $\Phi_s(k\Delta t)$ ，其中 $k \in \mathbb{R}$ ， $0 \leq k \leq \lfloor T/\Delta t \rfloor$ 。为了进一步加强安全，对最终轨迹周围具有固定半径的圆形管道进行碰撞检查，以保证轨迹和障碍物之间有足够的距离。优化程序在未检测到碰撞时停止。真实世界的实验环境与Teach-Repeat-Replan相同，使用Intel RealSense D435得到深度信息。此外，我们修改了英特尔RealSense的ROS驱动程序，每隔一帧启用激光发射器（感谢评论区指正）。这使得该设备可以在发射器的帮助下输出高质量的深度图像，以及不受激光干扰的双目图像。修改后的驱动程序也是开源的。

6.2 优化算法的比较

6.3 有/无ESDF的轨迹生成

6.4 多个规划器的比较

6.5 真实世界实验