Star-Convex Constrained Optimization for Visibility Planning with Application to Aerial Inspection

Tianyu Liu\textsuperscript{1,2,3}, Qianhao Wang\textsuperscript{1,2}, Xingguang Zhong\textsuperscript{1,2}, Zhepei Wang\textsuperscript{1,2}, Chao Xu\textsuperscript{1,2}, and Fei Gao\textsuperscript{1,2}

Abstract—The visible capability is critical in many robot applications, such as inspection and surveillance, etc. Without the assurance of the visibility to targets, some tasks eventually turn out to be unfinished or failed. In this paper, we propose visibility guaranteed planner by star-convex constrained optimization. The visible space is modeled as star convex polytope (SCP) by nature and is generated by finding the visible points directly on point cloud. By exploiting the properties of the SCP, the visibility constraint is formulated for trajectory optimization. The trajectory is confined in the safe and visible flight corridor which consists of convex polytopes and SCPs. We further make relaxation to the visibility constraints and transform the constrained trajectory optimization problem into an unconstrained one that can be reliably solved. To validate the capability of the proposed planner, we present the practical application in site inspection. The experimental results show that the method is efficient, scalable, and visibility guaranteed, presenting the prospect of application to various other applications in the future.

I. INTRODUCTION

In many applications, such as inspection and surveillance, enabling a drone to adjust its motion to keep interesting objects visible has high priority. Many tasks even put forward a strict demand on visibility. For instance, in substation inspection and factory security patrolling, specific positions must be repeatedly observed one by one in large-scale scenes. The tasks are regarded as unsuccessful or failed if any prescribed position is left unobserved. Therefore, visibility is a key constraint while designing a drone trajectory planner for these applications.

Despite the significance of visibility, most works [1]–[3] in the trajectory planning literature are not able to have a guarantee on it. Typically, they treat the visibility as a utility and optimize a handcrafted visibility cost along with other terms such as smoothness. However, such a formulation may trade-off visibility for a smoother motion, which results in soft visibility constrain. Another work [4] deterministically generates motion primitives and selects the best one among them. Although this method ensures viability in a resolution complete manner, it inherently suffers from the discretization error and the curse of dimension, which cannot generate an optimal trajectory with pleasing maneuverability.

To bridge the above gap, this paper proposes a planner that can efficiently generates a trajectory with visibility assurance. To generalize to various applications, we define the task representative point (TRP), which refer to the sites for inspection, the frontiers for the exploration, the places for surveillance, etc. Central to our approach is the visible space representation w.r.t. the TRPs and the corresponding constraint formulation. As we know, the line-of-sights from a TRP to space naturally form a star-shaped region. Based on this idea, we model the visible space as the star convex polytope (SCP), a compact and analytic representation. By utilizing the property of the constructed SCP, we formulate visibility constraint to facilitate star-convex constrained optimization.

In summary, the proposed planner optimizes trajectory in a safe and visible corridor (SVC) which encodes visibility and safety requirement. The SCPs, accounting for the visibility constraints, make one part of it. The SVC is constructed by connecting all the SCPs by convex polytopes. The whole process runs in three steps. Firstly, the global optimal tour (i.e., the visiting sequence of the SCPs) is found and refined on SCPs. Secondly, the kinodynamic A\textsuperscript{*} path searching is conducted to find a collision-free path. Finally, the corridor is constructed incrementally by connecting all the SCPs with sequences of overlapping convex polytopes utilizing the searched trajectory. With the constructed SVC, we follow the work of [5] to optimize the trajectory spatially and temporally. The visibility constraint is further relaxed
to convert the optimization problem into an unconstrained one that can be solved reliably and efficiently. To validate the planner, we apply it in the task of aerial inspection. Benchmark results show that our method is light-weighted, efficient, and scalable. To conclude, the contributions of this paper are as follows:

1) Introduce a new visible space representation the star-convex polytope (SCP) and propose to formulate the visibility constraint for star-convex constrained optimization.
2) Propose a visibility guaranteed planning framework, while retains the safety, feasibility, and energy efficiency of trajectory.
3) Validate the proposed method by implementing simulation and real-world experiments in aerial inspection.

II. RELATED WORK

A. Trajectory Planning with Visibility

Many works \[3\], \[6\] in trajectory planning design the visibility metric utilizing the minimum value of the Euclidean Signed Distance Field (ESDF) on the line between the TRPs and the robot. Since the metric is not differentiable, they use a sampling-based method to handle the metric in trajectory generation, which is time-consuming. Wang et al. \[2\] propose a differentiable metric and yet it lacks a strong guarantee on visibility because the trajectory optimization trades off many costs. Instead of explicitly optimize visibility, Zhou et al. \[7\] present a perception-aware strategy. Nevertheless, the task-specific method can hardly be extended to other scenarios. Zhou et al. \[8\] propose an efficient exploration framework that naturally adapts to inspection tasks, whereas they only consider visibility in the sampling-based front-end. In this paper, we efficiently extract the visible space by SCP to facilitate trajectory planning.

B. Trajectory Planning For Quadrotor

Trajectory planning for quadrotors can be categorized into the hard-constrained and soft-constrained approaches. The former formulates the trajectory generation as NLP to trade off several objectives, but they usually suffer from the issue of local minima \[9\]. By exploiting the properties of B-splines, Zhou et al. \[10\] propose a method but the construction of ESDF is time-consuming, especially for the large-scale trajectory planning. While an ESDF-free planner is proposed \[11\], but the trajectory generated highly rely on and limit to the collision-free guiding path. Hard-constrained methods usually formulate the problem as quadratic programming (QP) problem \[12\] with trajectory represented as piecewise polynomials. The safety can be ensured by extracting convex safe regions \[13\]. To obtain more reasonable time allocation, alternating minimization \[14\] and mixed integer-based \[15\] based approach are proposed. Recently, Wang et al. \[5\] proposed a spatial and temporal optimization-based framework, which efficiently handles a wide variant of constraints. We follow the work \[5\] for trajectory optimization in this paper.

III. PROBLEM STATEMENT

Consider a list of TRPs in 3D space \( \mathcal{C} = \{c_i \in \mathbb{R}^3 | 1 \leq i \leq N \} \). The robot starting from the position \( p_s \in \mathbb{R}^3 \) is expected to inspect all of the points in \( \mathcal{C} \) one by one and finally rest at the desired postion \( p_f \in \mathbb{R}^3 \). Commonly, the duration of inspection for each point is required to be last for at least a specific time \( T = \{t_i \in \mathbb{R} | 1 \leq i \leq N \} \). For an arbitrary point in \( \mathbb{R}^3 \), \( c_i \) is supposed to be visible to it if the line segment from the point to \( c_i \) is collision free. Denote \( S_i \subseteq \mathbb{R}^3 \) form the space where the point \( v_i \) is visible. Since the occlusion effect against obstacles i.e. the visibility is the focus of this paper, we make the following assumptions:

1) The sensor mounted on the robot has omnidirectional coverage, which is one kind of set up of UAVs and has certain research works \[16\].
2) The visibility condition of robot is satisfied only when the whole body of it is in the ball-shaped sensible regions around the points \( \mathcal{C} \).

IV. VISIBLE SPACE REPRESENTATION

For the TRPs \( \mathcal{C} \) to be seen, we need to obtain the space \( \mathcal{S} \) where the sites are visible to the drone. However, constructing a star-shape visible region on the point cloud map is non-trivial. Collision checking of the rays starting from the sites \( c_i \) to the space needs either frequent kd-tree queries or discretization of the space. Apparently, these kinds of straightforward solutions are arduous and time-consuming. Inspired by \[17\]–\[19\], we introduce a new method to construct visibility space represented by star convex polytope, with the emphasis on compactness and efficiency.

A. Star Convex Polytope Construction

In this paper, the obstacles are represented by point cloud map \( \mathcal{M}_g \) which is organized in k-d tree structure. Our method to construct star convex polytope on \( \mathcal{M}_g \) is composed of four steps: 1) points retrieve and augment, 2) points transformation, 3) convex hull construction, 4) inversion. The main idea of the method is to find the visible points by point transformation.

In order to construct the star-shaped region within a sphere boundary with radius \( R \) we retrieve the local point cloud \( \mathcal{M}_b \) surround the point \( c_i \) by the range query on \( \mathcal{M}_g \). In addition, augmented points, which are evenly sampled on the sphere boundary, are added to better facilitate the construction.

With the point set \( \mathcal{M}_b \) and center \( c_i \), we perform point transformation that flip all the points to outside of the sphere boundary. As shown in the Figure 2, the point \( x \) is transfer to \( \hat{x} \) along the ray \( \frac{c_i - x}{\|c_i - x\|} \). The corresponding function is suppose to be monotonically decreasing. Here, we simply use the ball flipping function with ball radius \( r \):

\[
\hat{x} = F(x) = x - c_i + 2(r - \|x - c_i\|) \frac{x - c_i}{\|x - c_i\|}. \tag{1}
\]

Then, we calculate the convex hull of the flipped points by the efficient convex hull algorithm \[20\]. Inherently, points that lie on the convex hull are the images of the visible points.
Similarly, the convex hull is the image of the underlying star-shaped boundary of visible space. Thus we can obtain the SCP by applying the inversion of (1) on the convex hull and denote it as \( S_i \). Moreover, the point that can be mapped outside the convex hull is bound to be visible by \( c_i \). The Point-In-SCP check can be performed by checking whether the flipping of the point is outside of the convex hull. This property of SCP will be employed in the following sections.

### B. Star-Convex Constrained Optimization

The visibility planning entails the study of the following optimization problem:

\[
\min_x \mathcal{J}(x), \quad \text{s.t.} \quad x \in S_i, \tag{2}
\]

where \( \mathcal{J}(x) \) is the user defined cost function. Suppose the SCP is closed by \( K \) faces in \( \mathbb{R}^3 \). Instead of considering it directly, the flipped convex polytope \( P_i \) is utilized. By the \( \mathcal{H} \)-representation of convex polytope, it can be defined as

\[
P_i = \{ x \in \mathbb{R}^3 | Ax \preceq b \}, \tag{3}
\]

where the matrix \( A = [n_1^T, \ldots, n_K^T] \in \mathbb{R}^{K \times 3} \) is build by the outer normal vectors of each face \( n_i, i = 1, \ldots, K \) and \( b = [n_1^T a_1, \ldots, n_K^T a_K] \in \mathbb{R}^K \) is formed by the arbitrary points \( a_i \) on each faces. By the property of SCP, the visibility constraint is equivuent to the ensurance that the flipped point \( \hat{x} \) is outside of \( P_i \), which is expressed as

\[
\Xi(\hat{x}) > d_{\text{min}}, \tag{4}
\]

where \( d_{\text{min}} \) is the user defined safe margin and \( \Xi(\cdot) \) is the signed distance function on \( P_i \). The signed distance equals to zero on the surface of the convex hull. The inside and the outside of it corresponding to the negative and positive euclidean distance respectively. To be more specific, the signed distance is defined as

\[
\Xi(\hat{x}) = \max \{ n_i^T (\hat{x} - a_i) \} | i = 1, 2, \ldots, K \}. \tag{5}
\]

Unexpectedly, the maximum function introduce the non-smooth gradient and keep it away form the efficient solution of the optimization with sophisticated solvers. As a matter of fact, the point visibility constraint can be enforced via smooth approximation of the maximum function. Inspired by [21], we employ the log-sum-exp function to make the approximation. Denote \( d_i = n_i^T (\hat{x} - a_i) \) for all \( i = 1, \ldots, K \). The (5) can be written as

\[
\Xi(\hat{x}) = \text{LSE}(d_1, \ldots, d_K) = \frac{1}{\alpha} \log(e^{\alpha d_1} + \cdots + e^{\alpha d_K}), \tag{6}
\]

where the \( \alpha \in \mathbb{R}^+ \) is an adjustable variable that can control the quality of the approximation, with \( \text{LSE}(d_1, d_2, \ldots, d_K) \rightarrow \max(d_1, d_2, \ldots, d_K) \) for \( \alpha \rightarrow +\infty \). Furthermore, we make relaxation of the original optimization problem (2) by constraint violation to convert it into an unconstraint problem:

\[
\min_x \mathcal{J}(x) + \mathcal{V}(\text{LSE}), \tag{7}
\]

where

\[
\mathcal{V}(\text{LSE}) = \lambda \max \{ \text{LSE}, 0 \}^3. \tag{8}
\]

The \( \lambda \in \mathbb{R}^+ \) is the penalty weight and the \( \text{LSE} \) stand for

\[
\text{LSE}(x|S_i) = d_{\text{min}} - \frac{1}{\alpha} \log(\sum_{i=1}^K e^{\alpha d_i}). \tag{9}
\]

Apparently, the violation term (8) preserve the \( C^2 \) condition, making the second order gradient attainable. Given the visible space \( S_i \), we can derive the gradient of \( \mathcal{V} \) w.r.t. \( x \) from (1), (8) and (9) and denote it by \( g_{\text{scp}} \). The gradient is zero when \( \text{LSE} \leq 0 \), and for \( \text{LSE} > 0 \), the gradient is given by

\[
g_{\text{scp}} = \frac{\partial \mathcal{V}}{\partial x} = 6\lambda \text{LSE}^2 \sum_{i=1}^K e^{\alpha d_i n_i} \frac{r}{\|x\|^3} (\|x\|^2 - xx^T - \|x\|^3 - \frac{2r}{2r}). \tag{10}
\]

We will employ the formation (7) for visibility planning in the following section.
Fig. 3. The whole pipeline is conducted in three steps: 1) route generation and refinement 2) path finding 3) SVC construction 4) trajectory optimization.

V. VISIBILITY GUARANTEED PLANNER

As is shown in the Figure 3, a complete pipeline of visibility guaranteed planner is presented in this section. Due to the differential flatness property of multicopters, we can optimize the trajectory in the space of the selected flat output \([p_x, p_y, p_z, \psi]\) (i.e. the translation of the center of mass and the Euler-yaw angle). To facilitate the trajectory optimization, the map \(\mathcal{M}_c\) is constructed by the one-to-eight cubic inflate of the points in the map of \(\mathcal{M}_g\). The \(\mathcal{M}_c\) encode the configuration space, and the SCP generated on it can be directly employed as the visibility constraint for trajectory optimization.

A. Route Generaton and Refinement

In order not to introduce the binary variable to the whole problem, a reasonable visiting sequence of the spots can be obtained in advance by solving the traveling salesman problem (TSP). Similar to \([8]\), we model it as a standard Asymmetric TSP (ATSP) that can be solved efficiently by LinKernighan heuristic (LKH) [22]. we further optimize route waypoints \(\{w_i \in \mathbb{R}^3\}_{i = 1, \ldots, N}\) on the SCPs to direct the robot for more efficient trajectory. The problem is formulated as finding the minimum of the sum of length on SCPs:

\[
\text{min}_{w_1, \ldots, w_N} \|p_s - w_1\| + \|w_N - p_f\| + \sum_{i=2}^{N} \|w_i - w_{i-1}\|, \\
\text{s.t.} \quad w_i \in \mathcal{S}_i, \quad \forall i = 1, \ldots, N.
\]

(11)

For simplification, \(w_0, w_{N+1}\) are alternatively used for \(p_s\) and \(p_f\) hereafter. According to (7), we further make a relaxation of (11) to convert it to an unconstrained NLP (nonlinear programming) with cost function

\[
J_w = \sum_{i=1}^{N} \sqrt{\|w_i - w_{i-1}\|^2 + \epsilon} + \lambda^T \sum_{i=1}^{N} \nu^I_{\text{LSE}}(w_i|\mathcal{S}_i),
\]

(12)

where the \(\Lambda = [\lambda_1, \ldots, \lambda_N]^T \in \mathbb{R}^N\) is the penalty weight vector and \(\epsilon\) is a small value number for \(C^2\) condition. By utilizing the previously-derived gradient \(g_{scp}\), the gradient propagation of \(J_w\) can be obtained for \(w_1, \ldots, w_N\):

\[
\frac{\partial J_w}{\partial w_i} = \frac{\|w_i - w_{i-1}\|}{\sqrt{\|w_i - w_{i-1}\|^2 + \epsilon}} - \frac{\|w_{i+1} - w_i\|}{\sqrt{\|w_{i+1} - w_i\|^2 + \epsilon}} + \lambda_i g_{scp}.
\]

(13)

Then, the route \(w_0 \rightarrow w_1 \rightarrow w_2 \rightarrow \cdots \rightarrow w_N \rightarrow w_{N+1}\) can be obtained by combining the problem of TSP with the optimization problem (11), which makes preparation for the corridor construction afterwards.

B. Safe and Visible Corridor Construction

The route generated is not collision free but provide promising flight directions. The route waypoints serve as local goals for kinodynamic A* to search for a collision free path. We convert the point cloud map to voxel map and perform the search on it, which can save orders of time.

Based on the searched path, the SVC can be constructed incrementally by connecting the SCPs by sequences of overlapping convex polytopes. For the convex polytope generation, we adopt the efficient method presented in [17] which directly makes modifications to SCP. Consequently, the elements of the corridor can be organized in a unified struct. The intersection between the path and convex polytope is calculated by recursively subdivide the Bézier form of the trajectory and checking the control points of it. For the SCP, the intersection between the path and it can be found via utilizing the property of the SCP. The convex polytope is built at the intersection until it reaches the next waypoint. Note that we add some augment points to separate the \(j^{th}\) and the \((j+2)^{th}\) convex polytopes.

C. Trajectory Optimization

Given the constructed SVC, the trajectory generation problem can be formulate as the following time-spatial optimization problem:

\[
\min_{\sigma(t)} \int_0^{T_\Sigma} \|\sigma^{(3)}(t)\|^2 dt + \rho T_\Sigma, \\
\text{s.t.} \quad [\sigma(0), \sigma^{(1)}(0), \sigma^{(2)}(0)] = [p_s, v_s, a_s],
\]

(14a)

\[
[\sigma(T_\Sigma), \sigma^{(1)}(T_\Sigma), \sigma^{(2)}(T_\Sigma)] = [p_f, v_f, a_f],
\]

(14b)

\[
\|\sigma^{(1)}(t)\| \leq v_m, \|\sigma^{(2)}(t)\| \leq a_m, \forall t \in [0, T_\Sigma],
\]

(14c)

\[
\sigma^I(t) \in \mathcal{S}_i, T_i > \tau_i, \forall i = 1, 2, \ldots, N,
\]

(14d)

where the \(\sigma(t) : \mathbb{R} \rightarrow \mathbb{R}^3\) is a polynomial spline over \([0, T_\Sigma]\) with time allocation \([T_1, T_2, \cdots, T_N]\) on SCPs, \(T_\Sigma\) the total time of \(\sigma(t)\), \(\rho\) the time regularization weight. The trajectory is constrained to be collision free, dynamic feasible, and visibility capable, which corresponding to the condition (14b), (14c) and (14d) respectively. Then, we denote by \(\mathcal{F}\) the resultant safe and visible corridor, \(v_m\) and \(a_m\) the dynamic limits, \(\sigma^I(t)\) the segment of \(\sigma(t)\) that assign to the \(j^{th}\) SCP.

To solve the optimization problem (14), we generally adopt the directly constructed minimum control trajectory
MINCO from [5]. Similar to [5], smooth maps are utilized to exactly eliminate spatial and time constraints. The dynamic constraint (14c) is transformed into a finite-dimensional one via integral of constraint violation. For brevity, we refer reader to [5] for more details.

For star-convex constraint in (14d), we make relaxation via integral of constraint violations. According to (7), we eliminate the constraint by defining the time integral penalty for visibility:

\[ I(S_i, \eta_i) = \frac{T_i}{\eta_i} \sum_{j=0}^{\eta_i} \mathcal{V}(\overline{LSE}(\sigma(\frac{T_j}{\eta_i}), S_i)), \quad (15) \]

where \( T_i \) is the time for the \( i \) segment of the trajectory, \( \eta_i \) controls the relative resolution of the quadrature. For the minimum time constraint in (14d), we take the decision variable mapping

\[ T_i = e^{\xi_i} + \tau_i, \quad (16) \]

to eliminate the constraint as well, where \( \xi = (\xi_1, \ldots, \xi_N) \) is \( C^\infty \) diffeomorphic to \( T = (T_1, \ldots, T_N) \). By incorporating (15) and (16) into the optimization framework [5], the optimization problem (14) can be transformed into the unconstrained control effort minimization problem which can be solved efficiently and reliably.

VI. APPLICATION ON AERIAL INSPECTION

Motivated by the need to regularly and regulate inspect sites [23], we test our planner on site inspection. The task requires that the drone can observe every spot for enough time while saving time and energy as much as possible to facilitate the mission.

A. Simulation and Benchmark Comparisons

We test the proposed method in a randomly generated environment consisting of pillar-shaped and ring-shaped obstacles. To demonstrate the superiority of our method in various environments with different scales,

- **Small**: 20 × 20 m, 15 pillars and 6 rings, 3 spots.
- **Medium**: 40 × 40 m, 60 pillars and 20 rings, 10 spots.
- **Large**: 80 × 80 m, 150 pillars and 60 rings, 20 spots.

We set the dynamic limits of drone as \( v_{\text{max}} = 4.0 \text{ m/s} \) and \( a_{\text{max}} = 6.0 \text{ m/s} \). All the simulations are conducted with a 2.6 GHz Intel i7-9750H processor.

In the implementation, we set \( R = 6.0 \text{ m} \) to confine the SCP in a ball, \( r = 20 \text{ m} \) for ball flipping. In the trajectory generation, we use, \( \rho = 150 \), \( \eta_l = 10 \). For the LSE function, we set \( \alpha = 100.0 \), which can make an approximation with the precision of 0.01. We benchmark the method with [8] which can naturally adapt to our inspection task. For a fair comparison, we do not consider the constraint (14d) because Zhou’s method cannot handle it. For Zhou’s method, we make a few modifications to fit into our application. Firstly, the space of every spot is discretized by \{0.5 rad, 0.5 rad, 0.5 rad\} in spherical coordinate system and the visible points are checked by raycasting. Secondly, the route is generated and refined by constructing a graph on visible points by euclidean distance instead of the path length searched by A*.

The Table I shows the statistic on the trajectory quality. The visible capability refers to the ratio of observed spots.

<table>
<thead>
<tr>
<th>Scene Scale</th>
<th>Method</th>
<th>Traj dur (s)</th>
<th>Int ((J^2))</th>
<th>Vis cap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>Zou et. al</td>
<td>6.9</td>
<td>546.5</td>
<td>80%</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>7.7</td>
<td>159.3</td>
<td>100%</td>
</tr>
<tr>
<td>Medium</td>
<td>Zou et. al</td>
<td>31.4</td>
<td>819.5</td>
<td>54%</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>32.8</td>
<td>371.2</td>
<td>100%</td>
</tr>
<tr>
<td>Large</td>
<td>Zou et. al</td>
<td>63.5</td>
<td>1063.9</td>
<td>37%</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>65.3</td>
<td>487.8</td>
<td>100%</td>
</tr>
</tbody>
</table>

Fig. 4. Simulation in large scene with scale 40 × 150 m. The scene is composed of 150 pillar-shaped obstacles and 60 ring-shaped ones. There are randomly generated 40 spots for inspection and the corresponding SCPs are shown in different colors. The trajectory is generated in 4.4s and is guaranteed to inspect all the sites.
the criterion of $\text{Int}(J^2)$ (time integral of squared jerk). This primarily benefits from the powerful trajectory optimization framework [5]. The optimized trajectory duration is higher than Zhou’s but still comparable to it. Without the hard visibility constraint, Zhou’s method tends to reduce the length of trajectory, which will reduce the execution time, as is shown in Figure 5.

The comparison of the computational time is shown in Figure 6. Our method is faster than Zhou’s by orders of magnitudes and is more reliable. Lacking a compact environment abstraction(e.g. SVC), the trajectory optimization time of Zhou’s takes almost 99% of the whole pipeline. The proposed method spends about 42% of total time for the generation of SCPs, route, and SVC, but they highly speed up trajectory optimization. As the problem scale increase, our method can still finish in seconds. A more large scale test of the proposed method is shown in Figure 4.

B. Real-World Experiment

We conduct real-world indoor experiment to validate the proposed method, as is shown in Figure 7. The upright cylindrical obstacles are the targets to be inspected. The map is pre-built using lidar by LIO-SAM [24] and the trajectory is planned offline. The quadrotor we used is equipped with an Intel Realsense D435 for state estimation and Insta 360 One X2\(^1\) for omnidirectional perception. The maximum velocity and acceleration are set as 1.5 m/s and 1.0 m/s\(^2\). The minimum inspection time for each object is set as 1.0s.

The test environment and the associate results are displayed in Figure 7. The quadrotor is able to inspect all the targets. Since the quadrotor is not necessary to be closest to the targets as long as they are visible, it decrease speed and inspect the target through the gap. The test show that the SCP can excavates almost all of the visible region and the formulated star-convex constrained optimization renders more reasonable trajectory for visibility planning.

VII. Conclusion

In this paper, we introduce a compact and efficient space representation the SCP and propose to formulate the visibility constraint for star-convex constrained optimization. By utilizing the SCP, we design a visibility guaranteed planning framework, while retains the safety, feasibility, and energy efficiency of trajectory. The experimental results show that the method is efficient, scalable, and visibility guaranteed.

The main limitation of our method the omnidirectional perception assumption of the sensor model. In the future, we will take limited FOV of sensors into consideration and plan the yaw angle in trajectory optimization.

\(^1\)https://www.insta360.com/
REFERENCES


