

# Motion Planning in 3D Environments Using Visibility Velocity Obstacles

Oren Gal, Yerach Doytsher

Mapping and Geo-information Engineering  
Technion - Israel Institute of Technology  
Haifa, Israel

e-mail: {orengal,doytsher}@technion.ac.il

**Abstract**—In this paper, we present a unique method combining visibility analysis in 3D environments with dynamic motion planning algorithm, named Visibility Velocity Obstacles (VVO). Our method is based on two major steps. The first step is based on analytic visibility boundaries calculation in 3D environments, taking into account sensors' capabilities including probabilistic consideration. In the second step, we generate VVO transferring visibility boundaries from the position space to the velocity space, for each object. Each VVO represents velocity's set of possible future collision and visibility boundaries. Based on our analysis in velocity space, we plan our trajectory by selecting robot's future velocity at each time step, tracking each specific target by considering visibility constraints as an integral part of the velocities space. We formulate the tracked target in the environment as part of our planner and include visibility analysis for the next time step as part of our planning in the same search space. We define visibility aspects as part of velocity space, where all the objects are modeled from the visibility point of view. We introduce a potential trajectory planner combining unified 3D visibility analysis for target tracking as part of dynamic motion planning.

**Keywords**- *Visibility; Motion planning, 3D; Urban environment; Spatial analysis.*

## I. INTRODUCTION

Trajectory planning has developed alongside the increasing numbers of Unmanned Aerial Vehicles (UAVs) all over the world, with a wide range of applications such as surveillance, information gathering, suppression of enemy defenses, air to air combat, mapping buildings and facilities, etc.

Most of these applications are involved in very complicated environments (e.g. urban), with complex terrain for civil and military domains [5]. With these growing needs, several basic capabilities must be achieved. One of these capabilities is the need to avoid obstacles such as buildings or other moving objects, while autonomously navigating in 3D urban environments.

Path planning problems have been extensively studied in the robotics community. These problems include finding a collision-free path in static or dynamic environments, i.e., environments having moving or static obstacles. Over the past twenty years, many kinds of path planning methods

have been proposed, such as starting roadmap, cell decomposition, and potential field [6].

In this paper, as far as we know for the first time, we present visibility aspects as part of velocity space, where all the objects are modeled from visibility point of view. We introduce potential trajectory planner combining unified 3D visibility analysis for target tracking as part of dynamic motion planning. In the first part, we formulate visibility boundaries problem and introduce analytic solution. Later on, we present the VVO method, demonstrated with visibility boundaries with cars, pedestrians and buildings visibility boundaries. In the last part, we suggest pursuer planner using VVO for UAV test case.

## II. RELATED WORK

Path planning becomes trajectory planning when a time dimension is added for dynamic obstacles [7][8]. Later on, a vehicle's dynamic and kinematic constraints have been taken into account, in a process called kinodynamic planning [9]. All of these methods focus solely on obstacle avoidance.

Trajectory planning for air traffic control and ground vehicles has been well studied [10], based on short path algorithms using 2D polygons, 3D surfaces [11]. UAVs navigation has also been explored with vision-based methods [12], with local planning or a predefined global path [13].

UAV path planning is different from simple robot path planning, due to the fact that a UAV cannot stop, and must maintain its velocity above the minimum, as well as not being able to make sharp turns.

The visibility problem has been extensively studied over the last twenty years, due to the importance of visibility in Geographic Information System (GIS) and Geomatics, computer graphics and computer vision, and robotics [1][3]. Accurate visibility computation in 3D environments is a very complicated task demanding a high computational effort, which could hardly have been done in a very short time using traditional well-known visibility methods [15]. The exact visibility methods are highly complex, and cannot be used for fast applications due to their long computation time. Previous research in visibility computation has been devoted to open environments using Digital Elevation Model (DEM) models, representing raster data in 2.5D (Polyhedral model), and do not address, or suggest solutions for, dense built-up

areas. Most of these works have focused on approximate visibility computation, enabling fast results using interpolations of visibility values between points, calculating point visibility with the Line of Sight (LOS) method [16]. Other fast algorithms are based on the conservative Potentially Visible Set (PVS) [17]. These methods are not always completely accurate, as they may render hidden objects' parts as visible due to various simplifications and heuristics.

### III. VISIBILITY BOUNDARIES ANALYSIS

We extend our previous work [2], developed for a fast and efficient visibility analysis for buildings in urban environments, and consider also a basic structure of cylinders, which allows us to model pedestrians and trees. Based on our probabilistic visibility computation of dynamic objects, we test the effect of these by using data gathered from Web-oriented GIS sources to update our estimation and prediction on these entities.

Dynamic objects such as moving cars and pedestrians, directly affect visibility in urban environments. Due to modeling limitations, these entities are usually neglected in spatial analysis aspects. We focus on three major dynamic objects in an urban case: moving cars and pedestrians. Each object is modeled with 3D boxes or 3D cylinders, which allow us to extend the use of our previous visibility analysis in urban environments presented for static objects [2].

#### 1) Moving Car

**3D Modeling:** As we mentioned earlier, Web-cameras in urban environments can record the moving cars at any specific time. Image sources such as web cameras, like other similar sensors sources, demand an additional stage of Automatic Target Detection (ATD) algorithms to extract these objects from the image [19]. In this research we do not focus on ATD, which must be implemented when shifting from the research described in the paper toward an applicable system.

The common car structure can be easily modeled by two 3D boxes, as can be seen in Figure 1, which is similar to the original car structure presented in Figure 1.

We define the Car Boundary Points (CBP) as the set of visible surfaces' boundary points of 3D boxes modeling the car presented in Figure 1. Each box is modeled as 3D cubic  $C_{car}(x, y, z)$  as presented extensively in [2] for a building model case.

**Car Boundary Points (CBP)** - we define CBP of the object  $i$  as a set of boundary points  $j = 1..N_{CBP\_bound}$  of the visible surfaces of the car object, from viewpoint  $V(x_0, y_0, z_0)$ , where the maximum surface's number is six and each surface defined by four points,  $N_{CBP\_bound} \leq 24$ , described in (1).



Figure 1. Car Modeling Using 3D Boxes

In Figure 2, the car is modeled by using two 3D boxes. Visible surfaces colored in red, the CBP marked with yellow points.

$$CBP_{i=1..N_{CBP\_bound}}(x_0, y_0, z_0) = \begin{bmatrix} x_1, y_1, z_1 \\ x_2, y_2, z_2 \\ \dots \\ x_{N_{CBP\_bound}}, y_{N_{CBP\_bound}}, z_{N_{CBP\_bound}} \end{bmatrix} \quad (1)$$

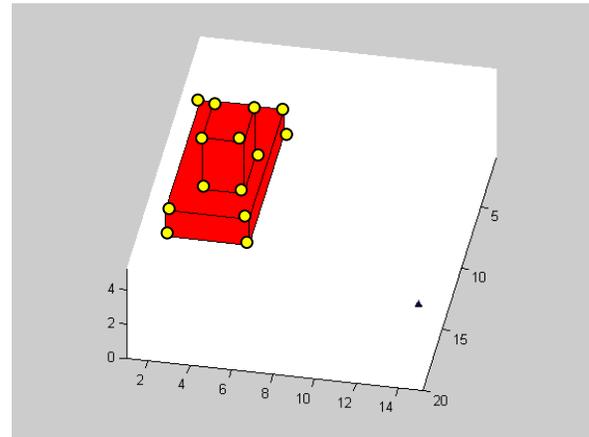


Figure 2. Modeling Car Using 3D Boxes (CBP Marked with Yellow Points)

#### Probabilistic Visibility Analysis

Visibility has been treated as a Boolean values. Due to incomplete information and the uncertainties of predicting the car's location at future times, visibility becomes much more complicated.

As it is well known from basic kinematics, CBP can be estimated in future time  $t + \Delta t$  as shown in (2):

$$CBP_i(t + \Delta t) = CBP_i(t) + V(t)\Delta t + \frac{A(t)\Delta t^2}{2} \quad (2)$$

Where  $V(t)$  is the car velocity vector  $V(t) = (v_x v_y)^T$ , and the acceleration vector  $A(t) = (a_x a_y)^T$ . Estimation of a car's location in the future based on a web camera is not a simple task. Driver behavior generates multi-decision

modeling, such as car-following behavior, gap acceptance behavior, or lane-change cases including traffic flow, speed etc.[20].

Our probabilistic car model is based on microscopic simulation models that were properly calibrated and validated using VISSIM simulation [20]. The average speed in urban environments is about 45 [km/hr], from a minimum of 40 [km/hr] up to a maximum of 50 [km/hr]. In the situation of a free driving case, which is the common mode in urban environments [21], the acceleration of a family car can change between 1 to 3.5 [m/sec<sup>2</sup>], and on average 2.5 [m/sec<sup>2</sup>].

As can be seen from several validations of car and driver estimation, velocity and acceleration are distributed as normal ones, and lead to normal location distribution in (3):

$$\begin{aligned} V(t) &\sim N(\mu = 45, \sigma^2 = 10) \\ A(t) &\sim N(\mu = 2.5, \sigma^2 = 1) \\ CBP(t + \Delta t) &\sim \sum N \end{aligned} \quad (3)$$

In time step t, where the car's location is taken from a Web-camera, visibility analysis from CBP(t) is an exact one, based on our previous visibility analysis [2], as seen in Figure 2. Visibility analysis becomes probabilistic for future time t + Δt, applying the same visibility analysis for CBP(t + Δt) presented in Figure 3.

In Figure 3, the car's location from a Web-camera appears in the bottom left side. For Δt = 2[sec], the car's location is marked by two 3D boxes, where CBP for each of them is the boundary of visible surfaces marked in red. The probability that the visible surfaces, which are bounded by CBP, will be visible in future time is based on the last update taken from the web application (depicted with arrows in Figure 3), computed by using two different random normal PDF values for V and A.

## 2) Pedestrians

**3D Modeling:** Pedestrian modeling can be done in high resolution, but due to ATD algorithms capabilities, pedestrians are usually bounded by a 3D cylinder and not as an exact detailed model [19]. For this reason, we model pedestrians as 3D cylinders, which is somewhat conservative but still applicable.

Pedestrian can be easily modeled by 3D cylinders, as seen in Figure 4 (marked in red), which is similar to the output from ATD methods tested on a Web-camera output recognizing walkers in urban environments.

We extend our previous visibility analysis concept [2] and include new objects modeled as cylinders as continuous curves parameterization, C<sub>Peds</sub>(x,y,z) in (4). Cylinder parameterization can be described as:

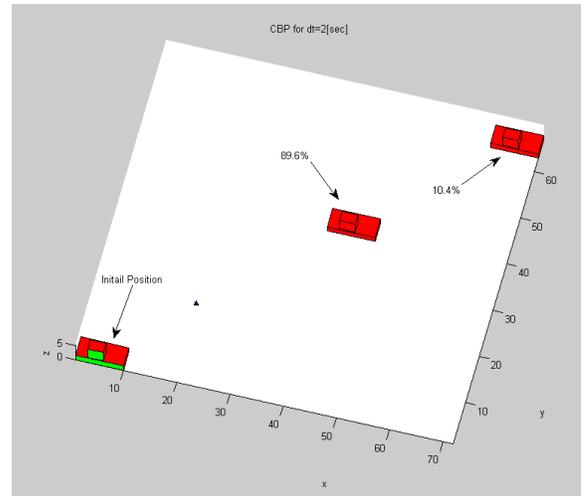


Figure 3. Probabilistic Visibility Analysis for CBP

$$\begin{aligned} C_{Peds}(x, y, z) &= \begin{pmatrix} r \sin(\theta) \\ r \cos(\theta) \\ c \end{pmatrix} \\ 0 &\leq \theta \leq 2\pi \\ c &= c + 1 \\ 0 &\leq c \leq h_{peds\_max} \end{aligned} \quad (4)$$

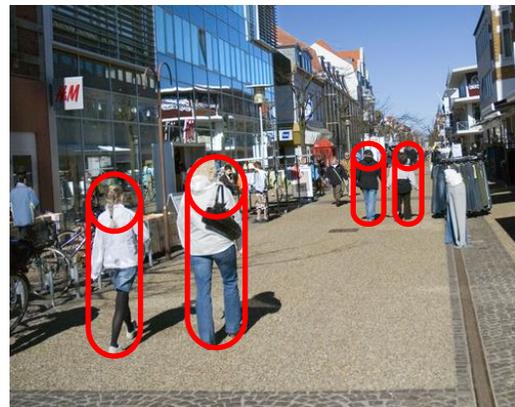


Figure 4. Modeling Pedestrians in Urban Scene Using Cylinders (Colored in Red)

We define the visibility problem in a 3D environment for more complex objects in (5):

$$C'(x, y)_{z_{const}} \times (C(x, y)_{z_{const}} - V(x_0, y_0, z_0)) = 0 \quad (5)$$

where 3D model parameterization is C(x,y)<sub>z=const</sub>, and the viewpoint is given as V(x<sub>0</sub>,y<sub>0</sub>,z<sub>0</sub>). Extending the 3D cubic parameterization, we also consider the cylinder case. As can be noted, these equations are not related to Z axis, and the visibility boundary points are the same for each x-y cylinder profile.

The visibility statement leads to a complex equation, which does not appear to be a simple computational task. This equation can be efficiently solved by finding where the equation changes its sign and crosses zero value; we used analytic solution to speed up computation time and to avoid numeric approximations. We generate two values of  $\theta$  generating two silhouette points in a very short time computation in (6). Based on an analytic solution to the cylinder case, a fast and exact analytic solution can be found for the visibility problem from a viewpoint.

$$\theta = \arctan \left( -\frac{-r - \left( -vy r + \sqrt{vx^4 - vx^2 r^2 + vy^2 vx^2} \right) vy}{vx^2 + vy^2} \right), \quad (6)$$

$$\left( -\frac{-vy r + \sqrt{vx^4 - vx^2 r^2 + vy^2 vx^2}}{vx^2 + vy^2} \right)$$

We define the solution presented above as x-y-z coordinates values for the cylinder case as **Pedestrian Boundary Points (PBP)**. PBP are the set of visible silhouette points for a 3D cylinder modeling the pedestrian in (7):

$$PBP_{i=1..N_{PBP\_bound}=2}(x_0, y_0, z_0) = \begin{bmatrix} x_1, y_1, z_1 \\ x_{N_{PBP\_bound}}, y_{N_{PBP\_bound}}, z_{N_{PBP\_bound}} \end{bmatrix} \quad (7)$$

#### IV. VISIBILITY VELOCITY OBSTACLES (VVO)

The visibility velocity obstacle represents the set of all velocities from a viewpoint, occluded with other objects in the environment. It essentially maps static and moving objects into the robot's velocity space considering visibility boundaries.

The VVO of an object with circular visibility boundary points such as the pedestrians case, PBP, that is moving at a constant velocity  $v_b$ , is a cone in the velocity space at point A. In Figure 5, the position space and velocity space of A are overlaid to illustrate the relationship between the two spaces. The VVO is generated by first constructing the Relative Velocity Cone (RVC) from A to the boundaries of the object, i.e., PBP, then translating RVC by  $v_b$ .

Each point in VVO represents a velocity vector that originates at A. Any velocity of A that penetrates VVO is an occluded velocity that based on the current situation, would result in an occlusion between A and the pedestrian at some future time. Figure 5 shows two velocities of A: one that penetrates VVO, hence, an occluded velocity, and one that does not. All velocities of A that are outside of VVO are visible from the current robot's position as the obstacle denotes as B, stays on its current course.

The visibility velocity obstacle thus allows determining if a given velocity is occluded, and suggesting possible changes to this velocity for better visibility. If PBP is known to move along a curved trajectory or at varying speeds, it

would be best represented by the nonlinear visibility velocity obstacle case discussed next.

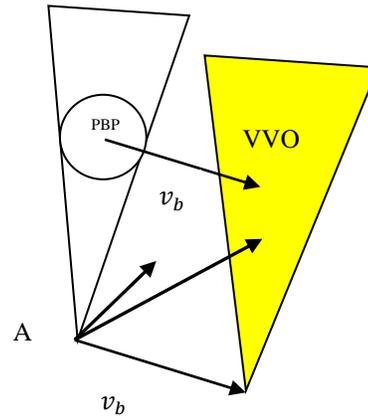


Figure 5. Visibility Velocity Obstacles

The VVO consists of all velocities of A at  $t_0$  predicting visibility's boundaries related to obstacles at the environment at any time  $t > t_0$ . Selecting a single velocity,  $v_a$ , at time  $t = t_0$  outside the VVO, guarantees visibility to this specific obstacle at time  $t$ . It is constructed as a union of its temporal elements,  $VVO(t)$ , which is the set of all absolute velocities of A,  $v_a$ , that would allow visibility at a specific time  $t$ .

Referring to Figure 6,  $v_a$  that would result in occlusion with point  $p$  in B at time  $t > t_0$ , expressed in a frame centered at  $A(t_0)$ , is simply in (8):

$$v_a = \frac{VBP_i}{t-t_0} \quad (8)$$

where  $r$  is the vector to point  $p$  in the blocker's fixed frame, and visibility boundaries denoted as Visibility Boundary Points (VBP). The set  $VVO(t)$  of all absolute velocities of A that would result in occlusion with any point in B at time  $t > t_0$  is thus in (9):

$$VVO(t) = \frac{VBP_i(t)}{t-t_0} \quad (9)$$

Clearly,  $VVO(t)$  is a scaled B for two dimensional case with circular object, located at a distance from A that is inversely proportional to time  $t$ . The entire VVO is the union of its temporal subsets from  $t_0$ , the current time, to some set future time horizon  $t_h$  in (10):

$$VVO(t) = \bigcup_{t=t_0}^{t_h} \frac{VBP_i(t)}{t-t_0} \quad (10)$$

The presented VVO generate a warped cone in a case of 2D circular object. If  $VBP(t)$  is bounded over  $t = (t_0, \infty)$ , then the apex of this cone is at  $A(t_0)$ . We extend our analysis to 3D general case, where the objects can be cubes, cylinders and circles. The mathematical analysis with visibility boundaries is based on VBP presented in the previous part

for different kind of objects such as buildings, cars and pedestrians.

We transform the visibility's boundaries into the velocity space, by moving the VBP to the velocity space, in the same analysis presented for 2D circle boundaries.

Following that, we present a 3D extension for VBP case, transformed to the velocity space.

Given two objects,  $VBP_1$ ,  $VBP_2$  will create a VVO representing  $VBP_2$  (and vice-versa) such that  $VBP_1$  wishes to choose a guaranteed collision-free velocity for the time interval  $\tau$ , and visibility boundary in velocity space.

In case of cars, buildings and pedestrians where visibility boundaries can be expressed by geometric operations of 3D boxes, analyzed in the same concept and formulation presented so far, as can be seen in Figure 6.

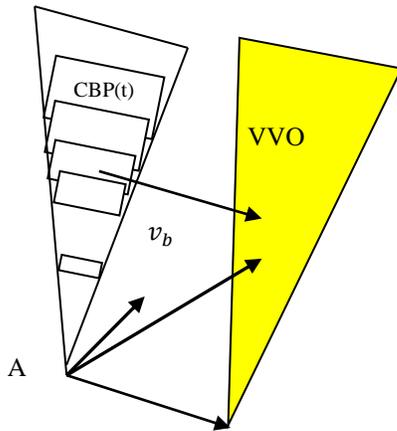


Figure 6. Visibility Velocity Obstacle for visibility boundaries consist of 3D boxes

## V. PURSUER PLANNER USING VVO

Our planner, similar to previous work [22] is a local one, generating one step ahead every time step reaching toward the goal, which is a depth first A\* search over a tree. We extend previous planners which take into account kinematic and dynamic constraints [9][14] and present a local planner for UAV as case study with these constraints, which for the first time generates fast and exact visible trajectories based on VVO, tracking after a target by choosing the optimal next action based on velocity estimation. The fast and efficient visibility analysis of our method allows us to generate the most visible trajectory from a start state  $q_{start}$  to the goal state  $q_{goal}$  in 3D urban environments, which can be extended to real performances in the future. We assume knowledge of the 3D urban environment model, and by using Visibility Velocity Obstacles (VVO) method to avoid occlusion, planner is based on exploring maximum visible node in the next time step and track a specific target.

### 1) Attainable Velocities

Based on the dynamic and kinematic constraints, UAVs velocities at the next time step are limited. At each time step during the trajectory planning, we map the AV, the velocities set at the next time step  $t + \tau$ , which generate the optimal trajectory, as it is well-known from Dubins theory [18].

We denote the allowable controls as  $u = (u_s, u_z, u_\phi)$  as  $U$ , where  $V \in U$ .

We denote the set of dynamic constraints bounding control's rate of change as  $\dot{u} = (\dot{u}_s, \dot{u}_z, \dot{u}_\phi) \in U'$ .

Considering the extremal controllers as part of the motion primitives of the trajectory cannot ensure time-optimal trajectory for Dubins airplane model [18], but is still a suitable heuristic based on time-optimal trajectories of Dubin - car and point mass models.

We calculate the next time step's feasible velocities  $\tilde{U}(t + \tau)$ , between  $(t, t + \tau)$  as shown in (11):

$$\tilde{U}(t + \tau) = U \cap \{u \mid u = u(t) \oplus \tau \cdot U'\} \quad (11)$$

Integrating  $\tilde{U}(t + \tau)$  with UAV model yields the next eight possible nodes for the following combinations in (12):

$$\tilde{U}(t + \tau) = \begin{pmatrix} \tilde{U}_s(t + \tau) \\ \tilde{U}_z(t + \tau) \\ \tilde{U}_\phi(t + \tau) \end{pmatrix} = \begin{pmatrix} u_s^{\min} u_s(t) + a_s \tau \\ -u_s^{\max} \tan \phi^{\max}, u_s(t) \tan u_\phi(t) + u_s^{\max} \tan a_\phi \\ u_z^{\max}, u_z(t) - a_z \tau \end{pmatrix} \quad (12)$$

At each time step, we explore the next eight AV at the next time step as part of our tree search, as explained in the next sub-section.

### 2) Tree Search

Our planner uses a depth first A\* search over a tree that expands over time to the goal. Each node  $(q, q)$ , where  $q = (x, y, z, \theta)$ , consist of the current UAVs position and velocity at the current time step. At each state, the planner computes the set of AV,  $\tilde{U}(t + \tau)$ , from the current UAV velocity,  $U(t)$ . We ensure the visibility of nodes by computing a set of Visibility Velocity Obstacles (VVO).

The search method is based on exploring nodes which are outside of VVO. The safe node with the lowest cost, which is the next most visible node, is explored in the next

time step. This is repeated while generating the most visible trajectory, as discussed in the next sub-section.

Attainable velocities profile is similar to a trunked cake slice, due to the Dubins airplane model with one time step integration ahead. Simple models attainable velocities, such as point mass, create rectangular profile [4].

### 3) Cost Function

Our search is guided by minimum invisible parts from viewpoint V to the 3D urban environment model, with minimal difference between robot's velocity  $v_a$  and tracked target  $v_{tck}$ .

The cost function is computed for each visible node  $(q, \dot{q}) \in VVO$ , i.e., node outside VVO, considering UAV velocities at the next time step in (13):

$$w(q(t + \tau)) = abs(v_a(q(t + \tau)) - v_{tck}(q(t + \tau))) \quad (13)$$

## VI. CONCLUSIONS

This paper proposes an online motion planning algorithm in 3D environments for tracking a target, taking into account visibility analysis. The planner is based on local search and includes dynamic and kinematic constraints as a complete part of the planner. Visibility boundaries which are based on analytic solution for several kinds of objects in 3D urban environments, also include uncertainty and probabilistic factors. Each VVO represents velocity's set of possible future collision and visibility boundaries. Based on our analysis in velocity space, we plan our trajectory by selecting future robot's velocity at each time step, tracking after specific target considering visibility constraints as integral part of the velocities space. We formulate the tracked target in the environment and include visibility analysis for the next time step as part of our planning in the same search space.

## REFERENCES

[1] G. Elber, R. Sayegh, G. Barequet, and R. Martin, "Two-Dimensional Visibility Charts for Continuous Curves," in Proc. Shape Modeling, MIT, Boston, USA, 2005, pp. 206-215.

[2] O. Gal, and Y. Doytsher, "Fast and Accurate Visibility Computation in a 3D Urban Environment," in Proc. of the Fourth International Conference on Advanced Geographic Information Systems, Applications, and Services, Valencia, Spain, 2012, pp. 105-110.

[3] O. Gal, and Y. Doytsher, "Fast Visibility Analysis in 3D Procedural Modeling Environments," in Proc. of the, 3rd International Conference on Computing for Geospatial Research and Applications, Washington DC, USA, 2012.

[4] P. Fiorini, and Z. Shiller, "Motion Planning in Dynamic Environments Using Velocity Obstacles," Int. J. Robot. Res. 17, 1998, pp. 760-772.

[5] Office of the Secretary of Defense, Unmanned Aerial Vehicles Roadmap, Tech. rep., December 2002.

[6] J.C. Latombe, "Robot Motion Planning," Kluwer Academic Press, 1990.

[7] M. Erdmann, and T. Lozano-Perez, "On Multiple Moving Objects," Algorithmica, 2, 1987, pp. 477-521.

[8] T. Fraichard, "Trajectory Planning in a Dynamic Workspace: A 'State-Time Space' Approach," Advanced Robotics, vol. 13, 1999, pp.75-94.

[9] S.M. LaValle, and J. Kuffner, Randomized Kinodynamic Planning. In Proc. IEEE Int. Conf. on Robotics and Automation, Detroit, MI, USA, 1999, pp. 473-479.

[10] Z.H. Mao, E. Feron, and K. Bilimoria, "Stability and Performance of Intersecting Aircraft Flows Under Decentralized Conflict Avoidance Rules," IEEE Transactions on Intelligent Transportation Systems, vol. 2, 2001, pp.101-109.

[11] J. Bellingham, A. Richards, and J. How, "Receding Horizon Control of Autonomous Aerial Vehicles," in Proceedings of the IEEE American Control Conference, Anchorage, AK, USA, 2002, pp. 3741-3746.

[12] B. Sinopoli, M. Micheli, G. Donata, and T. Koo, "Vision Based Navigation for an Unmanned Aerial Vehicle," in Proc. IEEE Int'l Conf. on Robotics and Automation, 2001.

[13] J. Sasiadek, and I. Duleba, "3D Local Trajectory Planner for UAV," Journal of Intelligent and Robotic Systems, vol. 2000, pp. 191-210.

[14] S.A. Bortoff, "Path Planning for UAVs," In Proc. of the American Control Conference, Chicago, IL, USA, 2000, pp. 364-368.

[15] H. Plantinga, and R. Dyer, "Visibility, Occlusion, and Aspect Graph," The International Journal of Computer Vision, vol. 5, 1990, pp. 137-160.

[16] Y. Doytsher, and B. Shmutter, "Digital Elevation Model of Dead Ground," Symposium on Mapping and Geographic Information Systems (Commission IV of the International Society for Photogrammetry and Remote Sensing), Athens, Georgia, USA, 1994.

[17] F. Durand, "3D Visibility: Analytical Study and Applications," PhD thesis, Universite Joseph Fourier, Grenoble, France, 1999.

[18] H. Chitsaz, and S.M. LaValle, "Time-Optimal Paths for a Dubins Airplane," in Proc. IEEE Conf. Decision and Control., USA, 2007, pp. 2379-2384.

[19] Y. Song: "The research of a new Auto Target Recognition directed Image compression," in 3th Int. Congress on Image and Signal Processing (CISP), 16-18 Oct, 2010, China.

[20] J. Archer: "Methods for the Assessment and Prediction of Traffic Safety at Urban Intersections and their Application in Micro-simulation Modeling," Centre for Traffic Simulation Research, CTR, Sweden. Technical Report, 2010.

[21] R. Wiedemann, and U. Reiter, "Microscopic Traffic Simulation: The Simulation System Mission, Background and Actual State," Project ICARUS (V1052), Final Report, Brussels CEC.2: Appendix A, 1992.

[22] O. Gal, and Y. Doytsher. "Patrolling Strategy Using Heterogeneous Multi Agents in Urban Environments Using Visibility Clustering," Journal of Unmanned System Technology, ISSN 2287-7320, 2016.